

**Proceedings of
Conference on Human Language Technology for
Development**

3-5 May 2011
Bibliotheca Alexandrina
Alexandria, Egypt

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

The conference is organized through the collaboration of PAN Localization (www.PANL10n.net) and ANLoc (<http://www.africanlocalisation.net>) Projects of IDRC (www.idrc.ca) and Bibliotheca Alexandrina (www.BibAlex.org).

Preface

Human Language Technology (HLT) is a growing field of research and development, converging multiple disciplines including computer science, engineering, linguistics, sociology and cognitive sciences, striving to develop a natural, easy and effective user interaction. HLT, including localization, is particularly relevant for addressing access to information by the disadvantaged communities, including the illiterate, the rural poor, and the physically challenged population, especially in the developing countries.

The Conference aims to promote interaction among researchers and professionals working on language technology, language computing industry, civil society engaged with deployment of language technology to end-users, and policy makers planning the use of HLT in national development projects. It aims to provide a single platform to engage these stakeholders in a dialogue over a wide range of relevant issues, to show-case state-of-practice in HLT and its use in development, and to identify needs and priorities of the end-users. It is hoped that the Conference will highlight HLTD challenges and viable solutions in the developing regions, especially in Asia and Africa.

We received 48 papers out of which 31 have been shortlisted for publication and presentation, after a double blind peer review process by the Technical Committee. The papers cover a variety of areas, including localization, development of linguistic resources, language processing and speech processing applications, and the challenges in the development and use of HLT. Seventeen papers are focused on Asian languages, while 14 focus on African languages, covering more than twenty languages from developing Asia and Africa.

We would like to thank the authors for their interest, the Technical Committee members for volunteering their time to review the papers, the Keynote Speakers and the Panelists. We would also like to thank our supporters and sponsor, including Asian Federation for Natural Language Processing, NECTEC, Thailand, and Arabize, Egypt. We acknowledge the support of the partners, including bibliotheca Alexandrina in Egypt, African Localization Network and Zuza Software in South Africa and PAN Localization at Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology in Pakistan. Finally, we are grateful to International Development Research Centre (IDRC) of Canada for their support through its Egypt and Singapore offices.

Organizing Committee
HLTD 2011

Organizing Committee

Dr. Adel El Zaim, IDRC, Middle East Office, Egypt (chair)
Dr. Ananya Raihan, D.NET, Bangladesh
Mr. Dwayne Bailey, Zuza Software Foundation, South Africa
Dr. Magdy Nagi, Bibliotheca Alexandrina, Egypt
Ms. Manal Amin, Arabize, Egypt
Ms. Maria Ng Lee Hoon, IDRC, SE and E Asia Office, Singapore
Dr. Peter Waiganjo Wagacha, Univ. of Nairobi, Kenya
Dr. Ruvan Weerasinghe, Univ. of Colombo School of Computing, Sri Lanka
Dr. Sarmad Hussain, Center for Language Engineering, KICS, UET, Pakistan

Technical Committee

Dr. Chafic Mokbel, Balamand University, Lebanon
Dr. Chai Wutiwiwatchai, NECTEC, Thailand
Mr. Donald Z. Osborn, African Network for Localization, USA
Dr. Friedel Wolff, Translate.org
Dr. Florence Tushabe, Makerere Univ., Uganda
Dr. Guy De Pauw, Univ. of Antwerp, Belgium
Dr. Hammam Riza, Agency for the Assessment and Application of Technology, Indonesia
Ms. Huda Sarfraz, Univ. of Engr. and Tech., Pakistan
Dr. Jackson Muhirwe, IT Dept., Lake Victoria Basin Commission, Kenya
Dr. Key-Sun Choi, Korean Advance Institute of Science and Technology, South Korea
Dr. Lamine Aouad, Univ. of Limerick, Ireland
Dr. Lisa Moore, Unicode Consortium, USA
Dr. Martin Benjamin, Kamusi Project International, Switzerland
Dr. Mekuria Fisseha, Makerere Univ., Uganda
Dr. Miriam Butt, Univ. of Konstanz, Germany
Dr. Mirna Adriani, Univ. of Indonesia
Dr. Mumit Khan, BRAC Univ., Bangladesh
Ms. Nayyara Karamat, Univ. of Eng. and Tech., Pakistan
Dr. Rajeev Sangal, International Institute of Information Technology, Hyderabad, India
Dr. Roni Rosenfield, Carnegie Mellon Univ., USA
Dr. Satoshi Nakamura, National Institute of Information and Communication Technology, Japan
Mr. Solomon Gizaw, Univ. of Limerick, Ireland
Dr. Steven Bird, Univ. of Melbourne, Australia
Dr. Tafseer Ahmed, Konstanz Univ., Germany
Dr. Tim Unwin, UNESCO Chair in ICT4D, Univ. of London, UK
Dr. Tunde Adegbola, African Languages Technology Initiative, Lagos, Nigeria
Dr. Virach Sornlertlamvanich, NECTEC, Thailand
Dr. Wanjiku Ng'ang'a, Univ. of Nairobi, Kenya

Keynote Addresses

The Language Technology Ecosystem, Richard L. Sites, Google
Arabic Language Processing and its Applications, Dr. Nabil Ali

Table of Contents

<i>Collation Weight Design for Myanmar Unicode Texts</i> Tin Htay Hlaing and Yoshiki Mikami	1
<i>Localising Microsoft Vista for Yoruba: Experience, Challenges and Future Direction</i> Tunde Adegbola	7
<i>Assessing Urdu Language Support on the Multilingual Web</i> Huda Sarfraz, Aniqā Dilawari and Sarmad Hussain	11
<i>LaoWS: Lao Word Segmentation Based on Conditional Random Fields</i> Sisouvanh Vanthanavong	21
<i>Burmese Phrase Segmentation</i> May Thu Win, Moet Moet Win, Moh Moh Than, Dr.Myint Myit Than and Dr.Khin Aye	27
<i>Dzongkha Text Corpus</i> Chungku, Jurmey Rabgay and Pema Choejey	34
<i>Towards a Sinhala Wordnet</i> Viraj Welgama, Dulip Lakmal Herath, Chamila Liyanage, Namal Udalamatta, Ruvan Weerasinghe and Tissa Jayawardhane	39
<i>CorpusCollie - A Web Corpus Mining Tool for Resource-Scarce Languages</i> Doris Hoogeveen and Guy De Pauw	44
<i>Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The Case of Amharic</i> Martha Yifiru Tachbelie, Solomon Teferra Abate and Laurent Besacier	50
<i>Language Resources for Mongolian</i> Purev Jaimaa and Altangerel Chagnaa	56
<i>Dzongkha Phonetic Set Description and Pronunciation Rules</i> Dechen Chhoeden, Uden Sherpa, Dawa Pemo and Pema Chhoejey	62
<i>Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System</i> Tadesse Anberbir, Tomio Takara, Michael Gasser and Kim Dong Yoon	68
<i>The Design of a Text Markup System for Yorùbá Text-to-Speech synthesis Applications</i> Odetunji Ajadi ODEJOBI	74
<i>Comparing Two Developmental Applications of Speech Technology</i> Aditi Sharma Grover and Etienne Barnard	81
<i>Phonetically Balanced Bangla Speech Corpus</i> S.M. Murtoza Habib, Firoj Alam, Rabia Sultana, Shammur Absar Chowdhury and Mumit Khan	87
<i>HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya</i> Michael Gasser	94
<i>Swahili Inflectional Morphology for the Grammatical Framework</i> Wanjiku Nganga	100

<i>Memory based Approach to Kikamba Name Entity Recognizer</i>	
Benson Nzioka Kituku, Peter W. Wagacha and Guy De Pauw	106
<i>Morphological Analysis and Machine Translation for Gikūyū</i>	
Kamau Chege, Wanjiku Ng'ang'a, Peter W. Wagacha, Guy De Pauw and Jayne Mutiga	112
<i>Understanding Natural Language through the UNL Grammar Workbench</i>	
Sameh Alansary, Magdy Nagi and Noha Adly	118
<i>Evaluation of crowdsourcing transcriptions for African languages</i>	
Hadrien Gelas, Solomon Teferra Abate, Laurent Besacier and François Pellegrino	128
<i>Development of an Open source Urdu screen Reader for Visually Impaired People</i>	
Madiha Ijaz and Qaiser Durrani	134
<i>Continuous Sinhala Speech Recognizer</i>	
Thilini Nadungodage and Ruvan Weerasinghe	141
<i>Dzongkha Text-to-Speech Synthesis System – Phase II</i>	
Dechen Chhoeden, Chungku , Chai Wutiwiwatchai, Ananlada Chotimongkol, Anocha Rugchat-jaroen and Ausdang Thangthai	148
<i>Bangla Text to Speech using Festival</i>	
Firoj Alam, S.M. Murtoza Habib and Mumit Khan	154
<i>A Corpus Linguistics-based Approach for Estimating Arabic Online Content</i>	
Anas Tawileh and Mansour Al Ghamdi	162
<i>Taxonomy of personalisation for Generating personalised content in Technical Support Forums</i>	
Solomon Gizaw and Jim Buckley	169
<i>Content independent open-source language teaching framework</i>	
Randil Pushpananda, Chamila Liyanage, Namal Udalamatta and Ruvan Weerasinghe	176
<i>English to Sinhala Machine Translation: Towards Better information access for Sri Lankans</i>	
Jeevanthi Uthpala Liyanapathirana and Ruvan Weerasinghe	182
<i>Strategies for Research Capacity Building in Local Language Computing: PAN Localization Project Case Study</i>	
Sana Shams and Sarmad Hussain	187
<i>Information Extraction and Opinion Organization for an e-Legislation Framework for the Philippine Senate</i>	
Allan Borra, Charibeth Cheng and Rachel Roxas	196

Conference Program

Tuesday May 3, 2011

(9:00 am) Opening Session

(9:40 am) Keynote Speech: The Language Technology Ecosystem, Richard L. Sites, Google

(10:40 am) Tea/Coffee

(11:00 am) Workshop 1: Leveraging the Web for Building Open Linguistic Data: Crowd-Sourcing Translation

(12:40 pm) Lunch

(2:00 pm) Workshop 2: Any language properly supported in CAT tools

(3:40 pm) Tea/Coffee

(4:00 pm) Workshop 3: Locale Workshop: Data Foundation for Language Infrastructure

Wednesday May 4, 2011

(9:00 am) Keynote Speech: Arabic Language Processing and its Applications, Dr. Nabil Ali

(10:40 am) Tea/Coffee

(11:00 am) Localization

11:00 *Collation Weight Design for Myanmar Unicode Texts*
Tin Htay Hlaing and Yoshiki Mikami

11:20 *Localising Microsoft Vista for Yoruba: Experience, Challenges and Future Direction*
Tunde Adegbola

11:40 *Assessing Urdu Language Support on the Multilingual Web*
Huda Sarfraz, Aniq Dilawari and Sarmad Hussain

12:00 *LaoWS: Lao Word Segmentation Based on Conditional Random Fields*
Sisouvanh Vanthanavong

Wednesday May 4, 2011 (continued)

12:20 *Burmese Phrase Segmentation*
May Thu Win, Moet Moet Win, Moh Moh Than, Dr.Myint Myit Than and Dr.Khin Aye

(12:40 pm) Lunch

(2:00 pm) Linguistic Resources

2:00 *Dzongkha Text Corpus*
Chungku, Jurmey Rabgay and Pema Choejey

2:20 *Towards a Sinhala Wordnet*
Viraj Welgama, Dulip Lakmal Herath, Chamila Liyanage, Namal Udalamatta, Ruvan Weerasinghe and Tissa Jayawardhane

2:40 *CorpusCollie - A Web Corpus Mining Tool for Resource-Scarce Languages*
Doris Hoogeveen and Guy De Pauw

3:00 *Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The Case of Amharic*
Martha Yifiru Tachbelie, Solomon Teferra Abate and Laurent Besacier

3:20 *Language Resources for Mongolian*
Purev Jaimaa and Altangerel Chagnaa

(2:00 pm) Speech Applications I

2:00 *Dzongkha Phonetic Set Description and Pronunciation Rules*
Dechen Chhoeden, Uden Sherpa, Dawa Pemo and Pema Chhoejey

2:20 *Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System*
Tadesse Anberbir, Tomio Takara, Michael Gasser and Kim Dong Yoon

2:40 *The Design of a Text Markup System for Yorùbá Text-to-Speech synthesis Applications*
Odetunji Ajadi ODEJOBI

3:00 *Comparing Two Developmental Applications of Speech Technology*
Aditi Sharma Grover and Etienne Barnard

Wednesday May 4, 2011 (continued)

3:20 *Phonetically Balanced Bangla Speech Corpus*
S.M. Murtoza Habib, Firoj Alam, Rabia Sultana, Shammur Absar Chowdhury and Mumit Khan

(3:40 pm) Tea/Coffee

(4:00 pm) Panel Discussion on Issues and Challenges in Local Language Computing: Perspectives from Asia and Africa

Thursday May 5, 2011

(9:00 am) Language Applications I

9:00 *HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya*
Michael Gasser

9:20 *Swahili Inflectional Morphology for the Grammatical Framework*
Wanjiku Nganga

9:40 *Memory based Approach to Kikamba Name Entity Recognizer*
Benson Nzioka Kituku, Peter W. Wagacha and Guy De Pauw

10:00 *Morphological Analysis and Machine Translation for Gĩkũyũ*
Kamau Chege, Wanjiku Ng'ang'a, Peter W. Wagacha, Guy De Pauw and Jayne Mutiga

10:20 *Understanding Natural Language through the UNL Grammar Workbench*
Sameh Alansary, Magdy Nagi and Noha Adly

Thursday May 5, 2011 (continued)

(10:40 am) Tea/Coffee

(11:00 am) Speech Applications II

- 11:00 *Evaluation of crowdsourcing transcriptions for African languages*
Hadrien Gelas, Solomon Teferra Abate, Laurent Besacier and François Pellegrino
- 11:20 *Development of an Open source Urdu screen Reader for Visually Impaired People*
Madiha Ijaz and Qaiser Durrani
- 11:40 *Continuous Sinhala Speech Recognizer*
Thilini Nadungodage and Ruvan Weerasinghe
- 12:00 *Dzongkha Text-to-Speech Synthesis System – Phase II*
Dechen Chhoeden, Chungku , Chai Wutiwiwatchai, Ananlada Chotimongkol, Anocha Rugchatjaroen and Ausdang Thangthai
- 12:20 *Bangla Text to Speech using Festival*
Firoj Alam, S.M. Murtoza Habib and Mumit Khan

(12:40 pm) Lunch

(2:00 pm) HLT Use

- 2:00 *A Corpus Linguistics-based Approach for Estimating Arabic Online Content*
Anas Tawileh and Mansour Al Ghamdi
- 2:20 *Taxonomy of personalisation for Generating personalised content in Technical Support Forums*
Solomon Gizaw and Jim Buckley
- 2:40 *Content independent open-source language teaching framework*
Randil Pushpananda, Chamila Liyanage, Namal Udalamatta and Ruvan Weerasinghe
- 3:00 *English to Sinhala Machine Translation: Towards Better information access for Sri Lankans*
Jeevanthi Uthpala Liyanapathirana and Ruvan Weerasinghe
- 3:20 *Strategies for Research Capacity Building in Local Language Computing: PAN Localization Project Case Study*
Sana Shams and Sarmad Hussain

Thursday May 5, 2011 (continued)

3.40 *Information Extraction and Opinion Organization for an e-Legislation Framework for the Philippine Senate*

Allan Borra, Charibeth Cheng and Rachel Roxas

(4:00 pm) Tea/Coffee

(4.20 pm) Panel Discussion on Wider Adoption of HLT for Development in Asia and Africa: The Way Forward

Collation Weight Design for Myanmar Unicode Texts

Tin Htay Hlaing

Nagaoka University of Technology
Nagaoka, JAPAN.

tinhtayhlaing@gmail.com

Yoshiki Mikami

Nagaoka University of Technology
Nagaoka, JAPAN.

mikami@kjs.nagaokaut.ac.jp

Abstract

This paper proposes collation weights for sorting Myanmar Unicode texts in conformity with *Unicode Collation Algorithm*. Our proposal is constructed based on the sorting order given in Myanmar Spelling Book which is also known as Myanmar Orthography. Myanmar language has complex linguistic features and needs preprocessing of texts before collation. Thus, we examine syllable structure, standard collation orders of Myanmar characters and then some preprocessing steps in detail. Furthermore, we introduced mathematical definitions such as *Complete Order Set (COSET)* and *Product of Order* to describe Myanmar sorting order in a formal way. This formal description gives clear and unambiguous understanding of Myanmar lexicographic order to foreigners as well as natives.

1 Aim of Research

Myanmar Language is the official language of Myanmar spoken by 32 million people as first language and as second language by some ethnic minorities of Myanmar.

However, awareness of Myanmar lexicographic sorting or collation order is lacking among the public. In Myanmar, people rarely look up a word in dictionary using Myanmar lexicographic order because most of the dictionaries published in Myanmar are English-to-Myanmar dictionaries. For example, Engineering dictionary, Medical dictionary and Computer dictionary use English lexicographic order. Likewise, in telephone directory and yellow pages in Web, the Myanmar names are Romanized and sorted according to the conventional Latin alphabet order.

Moreover, Microsoft's window operating system and office applications for desktop publishing are commonly used in Myanmar. But, until now, Microsoft applications such as Word and Excel simply sorts Myanmar character based

on binary code values and it does not meet Myanmar traditional sorting requirements.

Only a few professional linguists compile dictionaries but collation rules used by them are not understood easily by others. Lack of understanding of collation order in turn makes importance of collation unaware.

Consequently, these factors mentioned above act as a barrier to the implementations of Myanmar Language in database managements system and other ICT applications.

Therefore, we design collation weights for Myanmar Unicode texts and employed these collation weights in implementation of lexicographic sorting algorithm for Myanmar texts. Our algorithm is tested to sort the words using Myanmar Spelling Book order or Myanmar Orthography (Myanmar Language Commission, 2006) and proved its proper working. Also, we propose a formal description method for collation order using concepts of ordered set so that Myanmar sorting rules are well understood by both foreigners and natives.

2 Formal Description of Sorting Order

Standard sorting rules are often defined by national level language committee or are given in arrangement of words in the dictionaries. These rules are, however, complex, and difficult to understand well for those who are foreign to that language. Therefore, we introduced a formal description method for collation orders of syllabic scripts by employing a concept of Complete Order Set, COSET.

2.1 Definitions

Definition 1 Complete Order Set

In Set Theory, it is defined that a complete order is a binary relation (here denoted by infix \leq) on a set X . The relation is transitive, antisymmetric and total. A set equipped with a complete order is called a complete ordered set. If X is ordered

ခ, ဗ, ဏ, တ, ထ, ဒ, ဓ, န, ပ, ဖ, ဗ, ဘ, မ, ယ, ရ, လ, ဝ, သ, ဟ, ဋ, အ} becomes a COSET if normal consonant order is given. The order defined on C is written as (C, <).

3.2 Vowels

The set of Myanmar dependent vowel characters in Unicode contains 8 elements { ျ, ြ, ွ, ှ, ဿ, ျ, ြ, ွ, ှ, ဿ }. But some of them are used only in combined form such as

အ + ျ → အိ or အ + ွ → အီ or အ + ျ → အု and some characters which should be contracted before sorting such as အ + ျ + ျ → အိ. So, the vowel order is defined only on modified vowel set V` composed of 12 elements {အ, အာ, အိ, အီ, အု, အူ, အေ, အဲ, အော, အော်, အံ, အို} which it is written as (V`, <).

The vowel order is also not defined on independent vowel set I={အ, ဣ, ဤ, ဥ, ဦ, ဧ, ဩ, ဪ} because one letter ဦ needs to be normalized to arrange it in a complete order. Thus, vowel order is defined only on modified independent vowel set (Γ, <).

(V`, <) and (Γ, <) are isomorphic.

3.3 Medials

There are four basic medials characters M={ျ, ြ, ွ, ှ} which combine and produce total 11 medials. The set of those 11 medials have an order (M, <).

3.4 Finals or Ending Consonants

The set of finals F={ကံ, နံ, ဂံ, ဃံ, ငံ, စံ, ဆံ, ဇံ, ဈံ, ဋံ, ဌံ, ဍံ, ဎံ, ဏံ, တံ, ထံ, ဒံ, ဓံ, နံ, ပံ, ဖံ, ဗံ, ဘံ, မံ, ယံ, ရံ, လံ, ဝံ, သံ, ဟံ, ဣံ} having an order (F, <) is isomorphic to (C, <).

3.5 Diacritics

Diacritics alter the vowel sounds of accompanying consonants and they are used to indicate tone level. There are 2 diacritical marks { ျ, ြ } in Myanmar script and their order is (D, <).

3.6 Digits

Myanmar Language has 10 numerals N={ ၀, ၁, ၂, ၃, ၄, ၅, ၆, ၇, ၈, ၉ } with the order (N, <).

3.7 Myanmar Sign ASAT

When there is a consonant at the end of a syllable, it carries a visible mark call ASAT (ံ) to indicate that the inherent vowel is killed. It usually comes with consonants but sometimes comes with independent vowels.

4 Myanmar Syllable Structure in BNF

Most European languages have orders defined on letters, but those languages which use syllabic scripts, including Myanmar, have an order defined on a set of syllables. In Myanmar Language, syllable components such consonants(C), independent vowel(I) and digits(N) are standalone components while dependent vowels(V), medials(M), diacritics(D) and finals(F) are not. Among them, consonants can also act as nucleus of syllable so that other characters can attach to it in different combinations. The structure of Myanmar Syllable can be illustrated using BNF notation as S:= C{M}{V}{F}[D] | I[F] | N where { } means zero or more occurrences and [] means zero or one occurrence.

5 Comparison of Multilevel sorting in English and Myanmar

In English, the sorting process does on word level. After completing primary level comparison for a given word, the secondary level comparison is started for this word again. For Myanmar Language, the lexicographic order is defined by the product of five generic components, namely, consonant order (C, <), medial order (M, <), final order (F, <), vowel order (V, <) and diacritics order (D, <). Therefore Myanmar syllable order (S, <) is given by formula

$$(S, <)^M = (C, <) \times (M, <) \times (F, <) \times (V, <) \times (D, <)$$

Interesting to note here is the fact that the order (F, <) is considered before (V, <) while F comes after V in a coded string.

Thus, Myanmar sorting process does on syllable-wise behavior. For instance, a given word may contain one or more syllables. Sorting is done on first syllable and if there is no difference, the process will move to next syllable.

To sum up, firstly, one sort key is generated for one word in English while it is generated for each syllable in Myanmar. Secondly, multilevel sorting is done on word level in English but it is done within a syllable in Myanmar. Thirdly, in English, it needs whole word information because sorting process goes until it reaches end of word and then goes to next level. In contrast,

Myanmar sorting process goes until it reaches to last comparison level for one syllable and then it moves to next syllable. It means that Myanmar does not need whole word information if there is a difference before the final syllable.

6 Preprocessing of Texts

6.1 Syllabification

Myanmar is syllable based language and thus syllabification is necessary before collation. This process needs to return not only syllable boundary but also the type of each component within a syllable. Maung and Mikami (2008) showed syllable breaking algorithm with a complete set of syllabification rules.

6.2 Reordering

If Unicode encoding is followed, Myanmar characters are not stored in visual order. Myanmar vowels and consonant conjuncts are traditionally being typed in front of the consonants but we store Myanmar syllable components according to this order: <consonant> <medial> <vowel> <ending-consonant> <diacritic>. Therefore, no reordering is required for Myanmar Unicode texts.

6.3 Normalization

Normalization is required if a letter or ligature is encoded in composed form as well as decomposed form. One Myanmar character has multiple representations and thus normalization is required.

De-composed Form	Unicode for Decomposed forms	Equivalent Composed form	Unicode for Composed Form
ꠌ + ဝံ	1025 102E	ꠌ	1026

“Table 2. Normalization of a Vowel.”

6.4 Contractions

For some Myanmar dependent vowels, and medials, two or more characters clump together to form linguistic unit which has its own identity for collation. This group is treated similarly as a single character. These units may not be directly encoded in Unicode but are required to be

created from their constituent units which are encoded. This process is called *contraction* (Hussain and Darrani, 2004).

Glyph	Unicode for Contraction	Description
ꠌ + ဝံ ꠌ	1031+102C	Vowel sign E + AA
ꠌ + ဝံ ꠌ + ဝံ	1031+102C+103A	Vowel sign E+AA+ASAT
ꠌ + ဝံ	102D + 102F	Vowel sign I + UU

“Table 3. Vowel Contractions.”

Glyph	Unicode for Contraction	Description
ꠌ + ဝံ	103B + 103D	Consonant Sign Medial YA + WA
ꠌ + ဝံ	103C + 103D	Consonant Sign Medial RA + WA
ꠌ + ဝံ	103B + 103E	Consonant Sign Medial YA + HA
ꠌ + ဝံ	103C + 103E	Consonant Sign Medial RA + HA
ꠌ + ဝံ	103D + 103E	Consonant Sign Medial WA + HA
ꠌ + ဝံ + ဝံ	103B + 103D + 103E	Consonant Sign Medial YA+WA + HA
ꠌ + ဝံ ꠌ + ဝံ	103C + 103D + 103E	Consonant Sign Medial YA+WA + HA

“Table 4. Consonant Conjuncts Contractions.”

Similarly, we need contractions for ending consonants or finals which is a combination of consonants and Myanmar Sign ASAT. Some of them are shown in table below.

Glyph	Unicode for Contraction	Description
ꠌ + ဝံ	1000+103A	Letter KA + ASAT
ꠌ + ဝံ	1001+103A	Letter KHA+ ASAT
ꠌ + ဝံ	1002 + 103A	Letter GA + ASAT

“Table 5. Ending Consonant Contractions.”

7 Myanmar Collation

Myanmar Language presents some challenging scenarios for collation. Unicode Collation Algorithm (Davis and Whistler, 2010) is to be modified for Myanmar Collation. Because, in UCA, only one final sort key is generated for one word. But, one Myanmar word is divided into a sequence of syllables and sort key is generated at the syllable level. We use five levels of collation with consonants getting the primary weights and conjunct consonants having the secondary weights. At the tertiary and quaternary levels, ending consonants and vowels will be sorted respectively. Finally, the quinary level is used to sort diacritical marks. We ignore digits intentionally as there is no word with digits in dictionary. The collation levels and their respective weight ranges are depicted in table below.

Level	Components	Range
Primary	Consonants	02A1..02C2
Secondary	Consonant Conjuncts	005A..0064
Tertiary	Ending Consonants	0020..0050
Quaternary	Vowel	0010..001B
Quinary	Diacritics	0001..000D

“Table 6. Collation Levels and Weights Range for Myanmar.”

7.1 Collation Element Table for Myanmar

Some of the Unicode collation elements for Myanmar Language are given in Appendix A.

8 Conclusion

This paper proposes syllable-based multilevel collation for Myanmar Language. We also aimed to show how Unicode Collation Algorithm is employed for Myanmar words. We tested our algorithm to sort the words using Myanmar Orthography or Spelling Book Order and found that it works. But we have to do some more tests to handle loan syllable, kinzi, great SA and chained syllable so that we can produce more reliable evaluation. Myanmar language has some traditional writing styles such as consonant stacking eg. ဗုဒ္ဓ (Buddha), မန္တလေး (Mandalay, second capital of Myanmar), consonant repetition eg.

တက္ကသိုလ် (University), kinzi eg. အင်္ဂတေ (Cement), loan words eg. ဘတ်(စ်) (bus). Although we write using above traditional styles, we read them in a simple way, for example, တက္ကသိုလ် will be read as တက်+ က + သိုလ် by inserting invisible Virama sign automatically. Therefore, syllabification process has to provide syllable boundary according to the way we read. If syllable breaking function does not work well, it may affect our result.

References

Mark Davis and Ken Whistler. 2010. *Unicode Collation Algorithm, Version 6.0.0*.

Myanmar Language Commission. 2006. *Myanmar Orthography*, Third Edition, University Press, Yangon, Myanmar.

Robert M. Moll, Michael A. Arbib, and A.J. Kfoury. 1988. *An Introduction to Formal Language Theory*, Springer-Verlag, New York, USA.

Sarmad Hussain and Nadir Darrani. 2008. *A Study on Collation of Languages from Developing Asia*, National University of Computer and Emerging Sciences, Lahore, Parkistan.

Yoshiki Mikami, Shigeaki Kodama, and Wunna Ko Ko. 2009. *A proposal for formal description method of Collating order*, Workshop on NLP for Asian Languages, Tokyo.1-8.

Zin Maung Maung and Yoshiki Mikami. 2008. *A Rule-based Syllable Segmentation of Myanmar Texts*. Proceedings of the IJCNLP Workshop on NLP for Less Privileged Languages, Hyderabad, India, 51-58.

Appendix A. <Myanmar Collation Element Table>

Glyph	Unicode	Collation Elements	Unicode Name
<- Consonants ->			
က	1000	02A1 005A 0020 0010 0001	MYANMAR LETTER KA
ခ	1001	02A2 005A 0020 0010 0001	MYANMAR LETTER KHA
ဂ	1002	02A3 005A 0020 0010 0001	MYANMAR LETTER GA
.
အ	1021	02C2 005A 0020 0010 0001	MYANMAR LETTER A
<- Consonant Conjunct Signs or Medials ->			
ချ	103B	0000 005B 0020 0010 0001	MYANMAR CONSONANT SIGN MEDIAL YA
ြ	103C	0000 005C 0020 0010 0001	MYANMAR CONSONANT SIGN MEDIAL RA
့	103D	0000 005D 0020 0010 0001	MYANMAR CONSONANT SIGN MEDIAL WA
.
ချ့	103B103D103E	0000 0064 0020 0010 0001	MYANMAR CONSONANT SIGN MEDIAL YA+WA+HA
ြ့	103C103D103E	0000 0065 0020 0010 0001	MYANMAR CONSONANT SIGN MEDIAL RA+WA+HA
<-Independent Vowels ->			
ဣ	1023	0000 0000 0000 0012 0001	MYANMAR LETTER I
ဣါ	1024	0000 0000 0000 0013 0001	MYANMAR LETTER II
ဥ	1025	0000 0000 0000 0014 0001	MYANMAR LETTER U
ဥါ	1025102E	0000 0000 0000 0015 0001	MYANMAR LETTER U + MYANMAR LETTER II
ဦ	1026	0000 0000 0000 0015 0001	MYANMAR LETTER UU
.
ဪ	102A	0000 0000 0000 0019 0001	MYANMAR LETTER AU
<- Dependent Vowels ->			
ါ	102B	0000 0000 0000 0011 0001	MYANMAR VOWEL SIGN TALL AA
ာ	102C	0000 0000 0000 0011 0001	MYANMAR VOWEL SIGN AA
ိ	102D	0000 0000 0000 0012 0001	MYANMAR VOWEL SIGN I
.
ော	1031102C	0000 0000 0000 0018 0001	MYANMAR VOWEL SIGN E + AA
ောါ	1031102C103A	0000 0000 0000 0019 0001	MYANMAR VOWEL SING E+AA+ASAT
ံ	1036	0000 0000 0000 001A 0001	MYANMAR SIGN ANUSVARA
ိူ	102D102F	0000 0000 0000 001B 0001	MYANMAR VOWEL SING I + U
<- Diacritics ->			
့	1037	0000 0000 0000 0000 000C	MYANMAR SIGN DOT BELOW
း	1038	0000 0000 0000 0000 000D	MYANMAR SIGN VISARGA
<- Ending Consonants or Finals ->			
ကံ	1000103A	0000 0000 0021 0010 0001	MYANMAR LETTER KA + MYANMAR SIGN ASAT
.
အံ	1021103A	0000 0000 0042 0010 0001	MYANMAR LETTER A + MYANMAR SIGN ASAT

Localising for Yoruba: Experience, Challenges and Future Direction

Tunde Adegbola
African Languages Tech.
Init. (Alt-i), Ibadan, Nigeria

taintransit@hotmail.com

Kola Owolabi
University of Ibadan
Ibadan, Nigeria

kolawoleowolabi@yahoo.co.uk

Tunji Odejobi
Obafemi Awolowo
Univesity, Ile-Ife, Nigeria

oodejobi@yahoo.com

Abstract

This paper discusses the localisation of the MS Vista operating system for the Standard Yoruba language. A list of English lexical items used in MS Vista was provided and the task is to generate equivalent Yoruba terminologies. The Yoruba terminologies are required to convey the semantic connotation of the concepts presented in the MS Vista user interface. A number of techniques based on the linguistic structure of the Yoruba language were applied in generating translations and deriving equivalent lexical item words. The use of the glossary to develop the Yoruba version of MS Vista indicates that they capture the essence of the task required in the user interface. Although some user expressed reservation on non-technical aspect of the work, on the whole majority of the users expressed satisfaction on the outcome. We conclude from our experience that the task of localising for a language is a multi-disciplinary endeavour which will demand the expertise of Linguists and computer scientists.

1. Introduction

To improve the quality of human-computer interaction for Africans in the MS Vista environment, Microsoft Corporation extended its Local Language Program (LLP) to cover the localisation of Microsoft Windows operating system in various African languages. Yoruba is one of these languages. The African Languages Technology Initiative was invited into a partnership to moderate the development of Yoruba terminologies from a glossary of computer terminologies in English. A number of challenges were encountered in the process of developing the appropriate Yoruba glossary from the English glossary. This paper presents the efforts made in addressing these challenges and thereby suggests strategies that may be used to

facilitate similar future work in Yoruba and other African languages. In section 2 we describe the standard Yoruba Language briefly, with focus on its linguistic properties that are of interest in the morphology of lexical and related terms in the language. In Section 3, we discuss the methodology that underlies the technique that guided the generation of the lexical terms. Section 4 documents our experience as well as suggestions on extending the work to other African languages. Section 5 concludes this paper.

2. The standard Yoruba language

The Yoruba language comprises several dialects but this work is based on the Standard Yoruba (SY) language which is the language of trade, education, mass communication and general everyday interaction between Yoruba people, whatever their dialects of the language might be. SY is spoken by over 40 million people, mainly in West Africa. In Nigerian, it is spoken in Lagos, Osun, Ogun, Ondo, Oyo, Ekiti and Kwara, as well as some part of Kogi state. Yoruba is also one of the major languages spoken in Benin Republic and it is also spoken in some parts the Republic of Togo. In addition, the Yoruba language survived the transatlantic slave trade and is therefore used largely as a language of religion in a number of countries of the Americas such as Cuba and Brazil as well as some Caribbean countries.

Prior to the localisation of Microsoft Vista, Yoruba had never been widely used in the domain of modern technology in general, and computer technology in particular. However, there had been concern among

members of Egbe-Onimo-Eded-Yoruba (Yoruba Studies Association of Nigeria) about the need to widen the domains of use of the Yoruba language. This led to the development of *Ede-Iperi Yoruba* (Yoruba Metalanguage) Volume I in 1984 and Volume II in 1990. Between the dates of the publication of these two volumes of *Ede-Iperi Yoruba*, The National Language Centre of the Ministry of Education produced “A Vocabulary of Primary Science and Mathematics” in 9 Nigerian languages including Yoruba. *Ede-Iperi Yoruba* was addressed at the teaching of Yoruba through the medium of Yoruba in tertiary institutions while A Vocabulary of Primary Science and Mathematics was addressed at the teaching of mathematics and Science at the primary school level in Nigerian languages. Hence, even though these earlier efforts provided useful background materials, the localisation of MS Vista (or any other computer software for that matter) for Yoruba demanded more than mere translation of computer terms into Yoruba. The project necessarily demanded the creation of Yoruba equivalent terms, which involves application of scientific strategies and principles for technical-term creation.

3. Methodology

3.1 Strategies for Term compilation and terminography

Since the project was seen as one of term compilation and terminography rather than mere translation, basic strategies based on the following concepts used in term compilation and terminography were employed.

- a) Base forms
- b) Derivations
- c) Obtainable collocations

These strategies were adopted in the building of equivalent Yoruba terminologies from the glossary of terms supplied by Microsoft. Examples of their deployment are as follows:

Modulation as presented in the glossary is a concept that suggests changing of the behaviour something. Hence, to derive the Yoruba terminology for modulation, the base form; **modulate** was first derived as *yí (ìṣesí) padà*. From this base form, other English derivations from modulate were appropriately translated into Yoruba as follows:

Modulate	yí (ìṣesí) padà
Modulator	ayíṣesí-padà
Modulation	ìyíṣesí-padà
Demodulate	dá (ìṣesí) padà sípò
Demodulator	adaṣesí-padà
Demodulation	ìdáṣesí-padà
modulator/ demodulator	ayíṣesí-padà/ adaṣesí-padà

Other similar examples are:

Active	Aṣiṣé
Directory	àkójopò fáìlì
Domain	àgbègbè ìkápá
Services	àpèsè
Active document	àkòsílè aṣiṣé-lé-lórí
Active object	ohun aṣiṣé-lé-lórí
Active field	Ìdá rékòòdù aṣiṣé
Active window	wínnò aṣiṣé

Identify	ṣèdámò;(ṣè idámò)
Identified	àṣèdámò; àdámò
Identification	ìṣèdámò; idámò
Identifier	aṣèdámò
unique identifier	aṣèdámò àso
globally-unique identifier	aṣèdámò-àso káríayé

3.2 Formulation devices employed:

- a) Semantic extension

This involves extending the meanings of indigenous Yoruba words or items, e.g.

Buffer	Àká (original Yoruba meaning: barn, store)
Chart	àṭẹ (original Yoruba meaning: tray for displaying things)

b) Description

This involves describing English terms based on their salient attributes, such as functions/purpose, manner of application/production, appearance, behaviour and other noticeable peculiarities.

Function/purpose, e.g.

Certification Ìjèrìísí (lit. to bear witness to or provide evidence for something)

Manner of application or production, e.g.

Attachment Àsomó (lit. that which is attached to something else)

Appearance, e.g.

angle brackets àkámó onígún (lit. Brackets with angles)

Behaviour, e.g.

Brackets Àkámó (lit. That which encloses things)

c) Coinage

This involves inventing a word or phrase, e.g.

Alphabet Ábídí

Loan translation or calque, e.g.

d) Conditional Kání (lit. *if we say ...*)

This is a calque from the protasis of a Yoruba conditional sentence as opposed to the apodosis.

e) Borrowing

This involves adoption or borrowing of a word or linguistic expression. The borrowed words are integrated into the phonological structure of Yoruba, e.g.

Megabyte Mégábáìtì

f) Composition

This involves combining two or more Yoruba words or morphemes, e.g.

Active

Aṣiṣé (a-ṣe-iṣé
lit. Something that is working)

3.3 Derivational tactics

The most prominent tactic used is nominalisation (i.e. noun formation) via the highly productive morphological processes of, e.g.

a) pre-fixation

identifier Aṣèdámò (from ṣèdámò prefixed with a)

b) compounding

Index ètò ìtò kasí (from ètò and ìtòkasí implying a collection of indicators)

All these procedures were followed in order to ensure harmony and consistency in the formulation of Yoruba terms, thus enhancing prospects of transparency and acceptability of the created terms as well as convenience in pedagogy.

4. Some of the problems identified and our efforts at addressing them

The development of the Yoruba language in the direction of technological development has suffered over the years due to the use of English as the language of technology in Nigeria. Hence, many of the English terms used in MS Vista do not have corresponding Yoruba terms. It was necessary therefore to develop terms that can convey the meanings of the original English terms to the Yoruba user of MS Vista.

There were a number of challenges encountered in the project. One of the main challenges was that due to the use of English as the main language of education in Nigeria, most Yoruba computer scientists do not have sufficient competency in the Yoruba language to facilitate enough knowledge to produce appropriate terminologies. On the other hand, most Yoruba linguists do not have sufficient knowledge of computing to understand the

proper contexts of use of many words in the domain of computing. Hence, a linguist translated a data field in the sense of a playing field for data and a computer scientist translated stand-by mode as *moodu sitandibai*.

Another major challenge was the influence of Yoruba news casters who in course of their daily work have to translate English terms encountered in news items to Yoruba on the fly. These newscasters sometimes use inappropriate terms which usually stick. Hence, there was the challenge of changing such prevailing terms without causing confusion in the minds of computer users.

One of such is the use of the term *Èrọ ayára bí àṣá* for the computer in Yoruba news on radio and television. *Èrọ ayára bí àṣá* describes a machine that is as fast as the hawk. The order of speeds in computing and the flight of the hawk are different and so this term may not be as appropriate as *kòmputà* which is derived by borrowing and phonologising the original; computer.

5. Experience and suggestions on extending our work to other African languages

There were many other challenges faced during the project. Some of these stem from the low level of infrastructure development in Nigeria. Such facilities as constant electric power supply and Internet connectivity that are usually taken for granted in other parts of the world were not always available. Efforts to make up for these deficiencies resulted in increased cost of the project.

After the launching of the Yoruba version of Windows Vista, user response did not reflect the anticipated level of enthusiasm from Yoruba computer users. First, users might not have been sufficiently aware that the Yoruba component is available due to low publicity of the project. Second, most users who are aware of the product seem to prefer to continue working in the English Language environment in which they are already familiar.

User reviews were mixed. They vary from excitement through ambivalence to outright condemnation of the project.

Also of importance is the need to approach such localisation project from an interdisciplinary point of view. The project required expertise from three key areas: (i) linguistics, (ii) language technology, (iii) computer science.

5. Conclusion

In conclusion, the localisation project is a worthwhile project. The project is both viable and important. The curiosity aroused by the project has attracted some desired attention to the Yoruba language in and around West Africa. The documentation that accompanies the work will also serve as a valuable archival and research tool, if not now, in the future. Based on the experience in this project, we consider expanding the work to other software products such as word processors and spread sheets. The idea that anybody who wants to use the computer should be able to speak English is not tenable. There is no reason whatsoever why any Yoruba that need to get money out of an ATM machine must learn to speak English.

References

- Oladele Awobuluyi. 2008. *Eko Iseda-Oro Yoruba*. Montem Paperbacks, Akure.
- Kola Owolabi. 1989. *Ijinle Itupale Ede Yoruba: Fonoloji ati fonetiki*. IUP, Ibadan.
- Friedel Wolff. 2011. *Effective Change Through Localisation*. Translate.org.za.

Assessing Urdu Language Support on the Multilingual Web

Huda Sarfraz

Aniqa Dilawari

Sarmad Hussain

Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology

firstname.lastname@kics.edu.pk

Abstract

This paper presents an assessment of the support available for developing web content and services in Urdu language. The paper presents the methodology designed to conduct the assessment and presents the results in its context. The paper specifically analyzes Urdu language support from aspects of character set and encoding, font support, input methods, locale, web terminology, HTML, web technologies and advanced application support. The methodology proposed can also be extended and used to evaluate support of other languages for online publishing.

1 Introduction

The web is playing a pivotal role in bringing information to the populations around the world. Though a significant amount of the content on the web is in a few languages (Internet World Stats, 2010), the web has started becoming increasingly multilingual. With the linguistic and cultural diversity come specific requirements. For example, the HTML tags used in formatting online text are largely centric to Latin script and formatting, and would need to be revised to cater to other languages using other scripts. This is evident from the fact that underlining tag `<u>` causes text in languages using Indic scripts, which have a top-line instead of a base-line, to become unreadable (Lata, 2010) and is therefore not applicable.

As more languages come online, it is important to comprehensively assess the support provided for them on the multilingual web.

Urdu is one such language, with over 100 million speakers in Pakistan, India and other regions (Lewis, 2009). It is the national language of Pakistan and state language of India. Urdu uses the Nastalique writing system, which is highly cursive and context dependent, and is therefore very complex (Hussain, 2003; Wali et al., 2006).

Though basic support for publishing Urdu online exists, a comprehensive analysis of the requirements and existing support is needed so that gaps can be identified and addressed.

The need for local language computing was recognized and incorporated into the IT Policy of Pakistan in 2000. The Ministry of IT has been funding research and development in this area since then. Due to these and other similar efforts, Urdu in Pakistan falls within the category of moderately localized languages, with fairly active academic and research programs, fairly mature standards, basic applications and reasonable work in advanced applications (Hussain et al., 2008a).

This work defines a methodology to analyze and assess the support for Urdu language on the multilingual web, and presents the results obtained using the methodology. The work has been undertaken to develop a consistent framework for online publishing, especially for citizen services being provided by the government in Urdu in Pakistan.

Section 2 gives an overview of related work, followed by Section 3 which gives the assessment methodology and the results obtained for Urdu. The paper then concludes with some recommendations in Section 4.

2 Related Work

With an increasing focus across the globe on the creation of indigenous and local language content, efforts are also being made to enable support for multiple languages.

The World Wide Web Consortium (W3C) states one of its primary goals is to make the benefits of the web “available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability” (World Wide Web Consortium, 2011). The W3C Internationalization Activity in particular collaborates with W3C working groups and other organizations to enable web technologies for use with different languages, scripts and cultures.

As the online content becomes increasingly multilingual, there are multiple initiatives which are looking at existing and emerging challenges. The recent Multilingual Web project of W3C is one of the initiatives in this regard, organizing four public workshops for participants to learn about existing standards, to assess the current situation and to identify the gaps to be addressed.

In the issues being identified, Froumentin (2010) highlights web usage, and notes that 50% of the world population has access to the web but does not use it. One of the reasons cited is that native languages are not supported. Lata (2010) assesses web technology in the context of Indian languages. India has rich language diversity and Lata (2010) reports 122 major languages and 2371 dialects according to the census in 2001. The report presents a survey of multilingual web based services in India, looking at complexities in various scripts. Standardization issues are separated into three categories, with input, encoding and display issues making up the core category. The middleware category includes HTML, CSS, web accessibility and mobile web issues.

Constable and Nelson (2010) also notes character sets as posing problems in client-server interaction. It underlines the importance of “global-ready” HTML/CSS, and also formatting preferences specific to certain cultures.

Apart from the Multilingual Web Project, there are also other initiatives which have been focusing on language support on the web. The PAN Localization project is one such example, which focuses on assessing and developing the technology for enabling multilingual computing. Through the project, Hussain et al. (2005) report an analysis of the status of language computing support for 20 Asian languages. There has also been more detailed work on specific languages (e.g. PAN Cambodia 2007).

The W3C Internationalization Tag Set is an endeavor in the same direction. It is a W3C recommendation to help internationalize XML based content (Lieske and Sasaki, 2010).

3 Urdu Language Support Assessment

Assessing a language for online publishing would require to investigate its support at multiple levels. These include support in international standards, national recommendations based on international standards and frameworks, and availability of tools and applications for basic localization. Finally, for effective use, inter-

mediate and advance application support is also desired.

This section presents the analysis of Urdu language support on the web from nine different aspects including (i) character set and encoding, (ii) input methods, (iii) fonts, (iv) collation sequence, (v) locale, (vi) interface terminology, (vii) formatting based on HTML tag set, (viii) support through web technologies, and (ix) advanced application support. Each subsection briefly describes the methodology used to analyze support for each aspect and then presents the results for Urdu language.

3.1 Character Set and Encoding

Character set and encoding support for any language is the most basic level of support needed if the script of that language is to be represented on a computational platform. The character set for any language includes basic characters, digits, punctuation marks, currency symbols, special symbols (e.g. honorifics), diacritical marks and any other symbols used in conventional publishing.

Recently Unicode (Unicode 2010) has emerged as the most widely used international standard through which character sets for different languages are enabled on the web, and in the words of Tim Berners-Lee, is the path “to making text on the web truly global” (Ishida, 2010).

When a national character set and/or encoding mechanisms exist, the next step is to ensure that the character set is appropriately mapped on to the relevant section of Unicode standard. This is easier for scripts which are based on single or few language(s) (e.g. Khmer, Lao, Thai), but becomes increasingly essential and difficult for scripts which are used for writing many different languages (e.g. Arabic and Cyrillic scripts) because in such cases the relevant subset of characters within the same script have to be identified. This becomes even more difficult for some scripts as Unicode has redundancy and ambiguity due to backward compatibility and other reasons. As more than one Unicode can be used to represent a character, termed as variants, a mapping scheme also needs to be defined to reduce the redundancies. Unicode does not stipulate all such possibilities, and a national policy has to be set to interpret Unicode in the context.

In addition to mapping, normalization is needed when a character can be ambiguously represented either using a sequence of Unicode code points or a single code point. The Unicode standard defines normalization forms in order to

address this issue. However, the normalization set may not address all specific needs for a language and additional normalization tables need to be defined.

Other additional considerations include linguistic specification at national level and support at Unicode level for bidirectional behavior of a language (e.g. in Arabic script letters are written from right-to-left, but digits are written from left-to-right), character contraction (e.g. ‘ch’ is considered a single character in Slovak), case folding (Latin ‘a’ may be treated similar to ‘A’ in some contexts, e.g. to resolve domain names) and conjuncts (e.g. in Indic scripts, a consonant cluster may combine to give an alternate shape) should also be investigated for relevant languages and formalized. The Unicode standard recommends mechanisms to handle this, but it has to be determined whether sufficient support exists for this within Unicode and across various keyboards, fonts and applications.

Character Set and Encoding Assessment Methodology: The complete national Urdu character set for Pakistan, referred to as the *Urdu Zabta Takhti*, UZT 1.01, (Hussain et al., 2001) has already been defined and missing characters added to the Unicode standard (Hussain et al, 2002). However, there still no nationally accepted recommendation on which subset of Unicode from the Arabic script block (U+06XX) should be used for Urdu. Due to ambiguity in the Unicode this results in variation across different content developers on the selection of underlying Unicode characters. So a subset was defined in reference to the national character set. All normalization and mapping schemes to use this subset are also identified in the current work. The work also tested bidirectional features (Davis, 2009) for Urdu as supported by various applications.

Character Set and Encoding Assessment for Urdu: The recommended character sub-set for Urdu has been defined as part of the study and is available in Appendix A. The table shows only the characters required for Urdu within the Arabic code page. Some characters have been marked as variants (V). These are characters which are not part of the Urdu character set, but bear enough similarity with particular Urdu characters that they may be used interchangeably. These have been noted because they should be mapped to corresponding characters in core set for Urdu, if inadvertently used.

Normalization and mapping schemes needed to use with the encoding are also developed and are summarized in Table 1.

Table 1. Normalization composed and decomposed forms for Urdu.

Combining Mark	Composed Form	Decomposed Form	Unicode Normalized Form	
U+0653	U+0622	U+0627 U+0653	Defined	
U+0654	U+0623	U+0627 U+0654	Defined	
	U+0624	U+0648 U+0654	Defined	
		U+0649 U+0654	Not Defined	
		U+06CC U+0654	Not Defined	
	U+06C0		U+06D5 U+0654	Defined
			U+0647 U+0654	Not Defined
	U+06C2		U+06C1 U+0654	Defined
			U+0647 U+0654	Not Defined
	U+06D3		U+06D2 U+0654	Defined

The bidirectionality analysis showed that there are some shortcomings in terms of application support. Though most of the document processing applications support Urdu properly, the newer contexts are still behind in implementation. For example, the address bar for Google Chrome (version 6.0.472.63) does not support bidirectional text, but is supported by Internet Explorer and Firefox. This support is needed for properly displaying the recently announced Internationalized Domain Names (IDNs; Klensin 2010). This is illustrated in Figure 1 below, which shows the same string ووو۔لسانیات۔پاکستان in two different browsers.



Figure 1: Bidirectional text rendering inconsistencies in browsers for IDNs

3.2 Input Methods

Standardized input methods must be available in order to create web content and to enable users to interact with online systems and to create their own content, e.g. keyboards, keypads, on-screen keyboards, etc. All characters for the language must be supported by the keyboard layout. Any additional characters used, e.g. Latin ‘.’ and ‘@’ could also be supported to allow easier online access.

For many languages, a phonetic keyboard layout is possible, which allows for easier typing based on the sounds of the letters of QWERTY keyboard. Though these keyboards are normally ad hoc, they can allow for easier transition of users familiar with English keyboard to local languages and should be considered. Non-phonetic keyboards are usually based on the frequency of characters and have better efficiency, especially in the case of users who are accustomed to using them.

In deciding the keyboards to adopt, existing user base must be considered. Users who are used to an existing layout are usually reluctant to switching to a new layout even if it is more intuitive or efficient. Further, if additional characters are added to a keyboard, it is preferable for it to remain backward compatible, for adoption by existing users.

A key can represent different characters if used in conjunction with the Shift, Alt and Control keys. Though this increases the number of possibilities, increased combinations, or number of keyboard layers, significantly impact the usability of a keyboard.

Further, many languages may require additional rules, which take more than a single keystroke to generate context sensitive codes. The input method (along with the keyboard) should support such rules.

Input Methods Assessment Methodology:

The current work surveyed historical typewriter layouts, starting from the first layout standardized by the Pakistan Government in 1948. Six popularly used current keyboards are also analyzed as per the framework and two recommendations are made for use based on current use.

Input Methods Support Assessment for Urdu:

The CRULP 2-layer phonetic keyboard is recommended for users who are already familiar with English keyboard layouts. For new user, the Inpage Monotype layout, widely in use across the publishing industry, is recommended. However, these and other keyboards are missing ‘.’ and ‘@’ symbols used for web browsing and email, and thus they need to be updated. These keyboard layouts are shown in Figures 2a and b.



Figure 2a: Inpage Monotype layout



Figure 2b: CRULP Phonetic (2 Layered) layout

3.3 Web Fonts

A character is a logical entity that is visually represented through different fonts. The fonts must be based on Unicode encoding and in internationally acceptable formats, e.g. TrueType, OpenType, etc. for wider use online.

Web Fonts Assessment Methodology: In order to assess the support for Urdu, a detailed analysis of existing fonts was conducted. The fonts were analyzed in terms of the following aspects.

The character set for Urdu was sub-categorized into further levels: core, secondary, tertiary and variant. The core set is minimally needed to write Urdu. The secondary set includes characters which are used in Urdu but are not part of the core set. Without these, the text is still

readable, but with some difficulty. Tertiary characters were those that are used in Urdu, but their lack of support will only cause minor inconvenience. Variant characters are those that are not part of the Urdu set, but bear resemblance to core characters. If they are inadvertently used within the language, they must be supported in order to keep the text readable. This categorization was primarily done on a linguistic basis, however Google search hit counts for different character codes were used as a secondary quantitative source of evidence for this categorization. A support score was then calculated for each font being analyzed, using the scheme depicted in Table 2.

	Full Support Score	Partial Support Score	No Support Score
Primary Character	3	1.5	0
Secondary Character	2	1	0
Tertiary Character	1	0.5	0

Table 2: Scoring Scheme for Font Support

The scoring scheme is designed such that fewer points are deducted in case of lack of support of non-critical characters. Fonts that score higher provide better support for a particular language.

Font style and readability was analyzed with respect to different aspects like line height, curves and joins, kerning, hinting and other script specific features. User feedback was also taken into consideration for this purpose. Rendering speed for web fonts can critically affect the usability of web content. Font file size was used as an approximate measure for comparing font rendering speed. Licensing is another important aspect to consider while analyzing fonts. Fonts available under open licenses can be adjusted as per requirements by users and can be used in a wider variety of contexts.

Font embedding is becoming a critical aspect for enabling languages on the web. This is because computer systems are not usually shipped with Urdu fonts and a normal user may not know how to install such a font on his or her machine. Font embedding provides a convenient solution, where fonts are included in the content of the website and web pages are properly displayed even if the font is not installed on the user machine, as they are downloaded along with the webpage being accessed. Therefore, font embedding is also taken into account. The embedding

analysis is carried out for different combinations of browsers and operating systems.

Nastalique is the conventional writing style for Urdu, though Naskh style has also been in use (Wali et al. 2006), especially in case of typewriters. Therefore, five available Nastalique fonts and one popular Naskh font are analyzed.

Web Fonts Assessment for Urdu: Appendix B below gives a summary of the results. The character support percentage is calculated using the scheme in Table 2 divided by the maximum score possible. The font samples are shown in selected form in order to show box height which has a significant impact on font readability (Urdu Web Interface Guidelines & Conventions Project, 2008a and 2008b). Nafees Nastalique and Nafees Web Naskh are found to be the most suitable fonts for use on the web for Urdu.

3.4 Collation Sequence

Lexicographic sorting of textual data should also be supported if multilingual content is to be developed for the web. The collation sequence should be first linguistically determined and standardized at national level, and then incorporated technically. Unicode (2010a) provides specific mechanisms for implementing the collation sequence, using collation elements for the characters in a language (Davis 2010). A more detailed language based analysis is given by Hussain et al. (2008b).

Collation Sequence Assessment Methodology: Collation sequence should be assessed at two levels. First the sequence of characters must be defined at a linguistic level. At the second level, collation elements should be defined to realize the sequence computationally. Finally these should be made part of national and international standard, e.g. Common Locale Data Repository.

Collation Sequence Assessment for Urdu: Urdu character sequence was recently finalized by the National Language Authority based on earlier work by Urdu Dictionary Board and given in Figure 3 below. Corresponding collation elements have also been suggested by Hussain et al. (2008b).


```

/* empahsis tag */
em{
font-style:normal; /* font style
set to normal for text */
background:#000000; /* back-
ground color set to black */
color:#FFFFFF; /* text color set
to white */
font-weight:600;
font-size:24px;
text-decoration:none;
}

```

This effect of adjusting the em tag is shown in Figure 5, where the text is displayed in white with a black background.



Figure 5: Adjusted em tag applied to Urdu text

An additional tag is needed to support localized ordered lists for Urdu language. This exists at application level in some cases, however there is no support for it in the HTML standard.

Overall, the list of non-relevant tags for Urdu included the <i>, <rt>, <rp> and <ruby>. The list of tags that were adjusted for Urdu includes: <a>, <cite>, <ins>, <pre>, <textarea>, <address>, <code>, <kbd>, <samp>, , , , <select>, <button>, <input>, <option>, .

3.8 Web Technologies

Web technologies, in particular client, server and database technologies need to be tested to ensure that proper language support is available.

Web Technologies Assessment Methodology: The analysis for Urdu included server side scripting and database connectivity, in particular PHP with MySQL; ASP.net with Microsoft SQL Server. Display of local language strings in program editors and database fields are checked to ensure proper support, in addition to client end display. In addition, a form is designed to input and display information for testing purposes, shown in Figure 6. The default setting for fonts and dimensions are changed to adjust for Nastalique style.

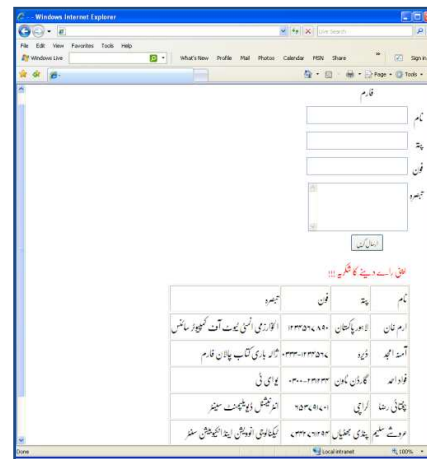


Figure 6: Form for web technology support assessment.

Web Technologies Assessment for Urdu: PHP/MySQL and ASP.net/Microsoft SQL Server were found to be generally supportive of Urdu and can be used for Urdu web applications. In both cases characters are also displayed properly within the database used.

3.9 Applications

Applications Assessment Methodology: A survey on advanced application support was also conducted to identify available applications which facilitate the uses online. This included typing tutors, spell checkers, online and desktop dictionaries, transliterators, machine translation systems, text to speech systems, automatic speech recognition systems and optical character recognition systems.

Advanced Applications Assessment for Urdu: No practically usable typing tutors are available for Urdu. Three pluggable spellcheckers are identified. Two in particular can be used in conjunction with Mozilla Firefox and OpenOffice.org. In addition, there are several useful online dictionaries and one desktop dictionary for Urdu. There are two transliteration utilities from ASCII to Urdu, both of which give reasonably robust transliteration results. Finally, there are two translation systems, one by Google and other by Ministry to IT; however neither is practically usable. Much more work needs to be done in this area.

4 Discussion and Conclusion

The analysis conducted shows that for Urdu language there are still some gaps in support.

Firstly, even though two fonts have been recommended through this study, the development of more optimal fonts is still needed. Secondly, some additional HTML support is needed. Thirdly, a lot more work is needed to provide adequate advanced application support.

Some of these gaps can be addressed through minor work-arounds, for example the adjustment of HTML tags through CSS. For other issues, updates are required in the standard, for example, support for localized ordered lists within HTML.

It is recommended that this analysis framework be used for other languages to assess support for online publishing and to identify and address gaps. These analyses can assist global efforts to provide support to create a truly multilingual web.

Acknowledgments

This work has been funded by the Punjab Information Technology Board of the Government of the Punjab, Pakistan.

References

- Constable, Peter and Nelson, Jan Anders. 2010. *Bridging Languages, Cultures and Technologies*. The Multilingual Web – Where are we (Workshop), Madrid, Spain.
- Davis, Mark. 2009. “Unicode Bidirectional Algorithm,” The Unicode Consortium. accessed from <http://www.unicode.org/reports/tr9/> on 22nd Sept. 2010.
- Davis, Mark and Whistler, Ken. 2010. Unicode Collation Algorithm 6.0.0. Unicode Consortium. Accessed from <http://unicode.org/reports/tr10/>.
- Froumentin, Max. 2010. *The Remaining 5 Billion: Why is Most of the World’s Population not Online and What Mobile Phones Can Do About It*. The Multilingual Web – Where are we (Workshop), Madrid, Spain.
- Hussain, Sarmad and Mohan, Ram. 2008a. Localization in Asia Pacific. In Digital Review of Asia Pacific 2007-2008. Orbicom and the International Development Research Center 2008.
- Hussain, Sarmad and Durrani, Nadir. 2008b. A Study on Collation of Languages from Developing Asia. PAN Localization Project, International Development Research Center, Canada.
- Hussain, Sarmad, Durrani, Nadir and Gul, Sana. 2005. PAN Localization Survey of Language Computing in Asia 2005. PAN Localization Project, International Development Research Center.
- Hussain, Sarmad. 2003. www.LICT4D.asia/Fonts/Nafees_Nastalique, in the Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore.
- Hussain, Sarmad and Zia, Khaver. 2002. “Proposal to Add Marks and Digits in Arabic Code Block (for Urdu)”, in the Proceedings of 42nd Meeting of ISO/IEC JTC1/SC2/WG2, Dublin, Ireland.
- Hussain, Sarmad and Afzal, Muhammad. 2001. Urdu Computing Standards: UZT 1.01, in Proceedings of the IEEE International Multi-Topic Conference, 2001, Lahore.
- Internet World Stats 2010, www.internetworldstats.com
- Ishida, Richard. 2010. *The Multilingual Web: Latest Developments at the W3C/IETF*. Workshop on The Multilingual Web – Where are we, Madrid, Spain.
- Klensin, John. 2010. RFC 5891: Internationalized Domain Names in Applications (IDNA): Protocol. Internet Engineering Task Force. Accessed from <http://tools.ietf.org/html/rfc5891>.
- Lata, Sawaran. 2010. *Challenges of Multilingual Web in India: Technology Development and Standardization Perspective*. The Multilingual Web – Where are we (Workshop), Madrid, Spain.
- Lewis, M. Paul (ed.). 2009. *Ethnologue: Languages of the World*, Sixteenth edition. Dallas, Tex.: SIL International. Online version: www.ethnologue.com
- Lieske, Christian and Sasaki, Felix. 2010. *WC3 Internationalization Tag Set*. The Multilingual Web – Where are we (Workshop), Madrid, Spain.
- PAN Cambodia 2007. Gap Analysis of HTML for Khmer, http://panl10n.net/english/Outputs%20Phase%202/CCs/Cambodia/MoEYS/Papers/2007/0701/HTML_Standard_for_EnglishAndKhmer.pdf. PAN Localization Project, Cambodia.
- Unicode 2010a. Unicode 5.0. 5th Edition. Addison-Wesley Professional, USA.
- Urdu Web Interface Guidelines & Conventions Project. 2008a. “Urdu Web Font Evaluation Criteria”, University of Management and Technology. Unpublished report.
- Urdu Web Interface Guidelines & Conventions Project. 2008b. “Usability Testing Report of Urdu Web Fonts”, University of Management and Technology. Unpublished report.
- Wali, Amir and Hussain, Sarmad. 2006. *Context Sensitive Shape-Substitution in Nastaliq Writing System: Analysis and Formulation*. In the Proceedings of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 06).

Appendix B. Font Analysis Results for Urdu

Font	Sample	Character support	Style and readability	Font file size	Embedding	License
Nafees Nastalique	چچا چھکن نے تصویر ناگی	96%	High	388KB	Embeddable with some minor issues	Open
Alvi Nastalique	چچا چھکن نے تصویر ناگی	98%	High	9MB	Embeddable but impractical	Free, for personal use only
Fajer Noori Nastalique	چچا چھکن نے تصویر ناگی	93%	Low	255KB	Embeddable	Information not available
Jameel Noori Nastalique	چچا چھکن نے تصویر ناگی	98%	High	13MB	Embeddable but impractical	Free for use
Nafees Web Naskh	چچا چھکن نے تصویر ناگی	99%	Low	124KB	Embeddable	Open
Pak Nastalique	چچا چھکن نے تصویر ناگی	96%	Low	167KB	Embeddable	Free for use

LaoWS: Lao Word Segmentation Based on Conditional Random Fields

Sisouvanh Vanthanavong
Information Technology Research
Institute, NAST, Lao PDR
sisouvanh@nast.gov.la

Choochart Haruechaiyasak
Human Language Technology laboratory,
NECTEC, Thailand
choochart.haruechaiyasak@nectec.or.th

Abstract

In this paper, we propose a word segmentation model based on the Conditional Random Fields (CRFs) algorithm for Lao language called Lao Word Segmentation (LaoWS). LaoWS is trained from a given corpus called Lao text corpus (LaoCORPUS). LaoCORPUS contains approximately 100,000 manually tagged words from both formal and informal written Lao languages. Using the CRFs algorithm, the problem of word segmentation can be formulated as a sequential labeling task in each character labeled with one of two following classes: word-beginning (B) and intra-word (I) characters. To train the model, we design the feature set based on the character tagged corpus, example, by applying all possible characters as features. The experimental results showed that the performance under the F-measure is equal to 79.36% compared to 72.39% by using the dictionary-based approach. As well as, using the CRFs approach, the model can segment name entities better than the dictionary-based approach.

Index Terms— *Lao Word segmentation, Tokenization, Conditional Random Fields*

1. Introduction

Especially the development of localization, Lao language is one of many languages in South East Asia countries which does not have any white space between syllables and words that called mono-syllable language. By the way, Lao language is still lacking a standard of lexical and Dictionary Base for language development of information technology field. However, this paper will technically present the key point of word segmentation task in text allocation analysis depending on a given corpus.

For many years, the researchers have been developing word segmentation in many difference languages by using Machine Learning Based and Dictionary Base. The main purpose of machine learning base is an independence of dictionary that opposites of Dictionary Based Approach. The unknown words and name entity can be solved by a model classification of machine learning approach. For example, neural network, decision tree, conditional random fields (CRFs) and etc.

Recently, many organizations and private sectors in Laos try to develop Lao language especially the information technology fields (text processing). LaoScript¹ for Windows, it has been developed for many years using in Microsoft Office and Text editor. Otherwise, PANL10N² project is one of the sectors to research and develop about natural language processing in Asian language. The main purposed of this paper is to produce machine learning base by a model classification for word segmentation task in the specific area into text processing from a given corpus and evaluation the proposed method by the performance of F-measure. To remain this paper is established as follows, the next section is about previous work in word segmentation. In section 3, a brief of CRFs algorithm, section 4 the main propose of word segmentation, and to be more practical, there will be the experiments and results in section 5, eventually section 6, will be the conclusion of this research.

2. Related Work

Recent year, Lao localization development has been analyzed in many fields, especially text

¹ <http://www.laoscript.net>

² <http://www.panl10n.net>

processing such as: line breaking system, convert fonts, spelling check and etc. For, i.e., Line breaking is very important for justification in Lao language, according to Lao line breaking (Phissamay *et al*, 2004) that created a new rule and condition for solving the problems of syllable breaking system in text editor, which's given the best performance up to 98%. However, the difficulty to technically improve from syllable breaking to word breaking system in text processing is about lacking of the lexical corpus and dictionary standardization.

Fortunately, Lao and Thai have a very similar language by spoken and writing system. Years ago, Thai language (Kruengkrai and Isahara, 2006; Theeramunkong and Usanavasin, 2001; Khankasikam and Muansuwan, 2005) was researched from syllable segmentation to word segmentation task using a rule-based system of language models and lexical semantic approaches, the decision tree model solves the word segmentation without a dictionary based, this result is given the accuracy approximately 70%, for the Dictionary Based Method gives the high accuracy approximately 95% with a context dependence.

Thai word segmentation approach (Haruechaiyasak *et al*, 2008; Thai Lexeme Tokenization. Online: 2010) produced the two different algorithms such as the Dictionary Bases (DCB) and Machine Learning Base (MLB). Normally, DCB approach (Sornil and Chaiwanarom, 2004) uses the Longest Matching (LM) technique to consider about information segmentation with the long word; Maximal Matching (MM) uses the existing word in the Dictionary base by selecting the segmented series that yields the minimum number of word taken. Otherwise this research described the experiments of the n-grams model of different character types from Thai text corpus by using Machine Learning Approach such as Naive Bayes (NB), Decision tree, Support Vector Machine (SVM), and CRFs. The result of this research selects the CRFs algorithm as the best way to detect Thai word boundary in machine learning based, with the precision and recall of 95.79% and 94.98% respectively.

Therefore, this research uses the CRFs algorithm in the challenging task of Lao word segmentation development.

3. CRFs Algorithm

In a CRF algorithm (Wallach, 2004; Lafferty *et al*, 2001; Alba *et al*, 2006) by a chain-structured model depending on each label sequence contains beginning and ending states respectively

y_0 and y_{n+1} , the probability of label sequence y that given an observation sequence x is $p(y | x, \phi)$ maybe efficiently computed matrices. We define y and y' that are the label sequences of an alphabet Y , a set of $n+1$ matrix $\{M_i(x) | i = 1, \dots, n+1\}$, where each $M_i(x)$ is a $|Y \times Y|$ matrix elements may be written

$$M_i(y', y | x) = \exp\left(\sum_j \phi_j f_j(y', y, x, i)\right).$$

The un-normalized probability of label sequence y given observation sequence x that considers the product of the matrix elements of the form of these label sequences:

$$p(y | x, \phi) = \frac{1}{Z(x)} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i | x).$$

Similarly, the normalization factor above $Z(x)$ is given by the (*start* and *end*) entry of the product of all $n+1$ and $M_i(x)$ matrices as follows:

$$\begin{aligned} Z(x) &= [M_1(x)M_2(x)\dots M_{n+1}(x)]_{start,end} \\ &= \left[\prod_{i=1}^{n+1} M_i(x)\right]_{start,end} \end{aligned}$$

To assume that the conditional probability of a label sequence y might be written as:

$$p(y | x, \phi) = \frac{\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i | x)}{\left(\prod_{i=1}^{n+1} M_i(x)\right)_{start,end}}$$

(LaoCORPUS). We will use the LaoCORPUS as a text file to create Lao tagged corpus.

Third, CRFs model needs a training set to perform segmentation task of text information, a training set is used as data set for CRFs learning model which our data set perform types feature based on character sets, CRFs model is predicted the possible character features from text information by using conditional probability of Hidden Markov Model (HMM) (Rabiner and Juang, 1986; Sarawagi and Cohen, 2005) and Max-entropy HMM combination (Zhao *et al*, 2007).

Finally, Implementing features model, beside the rule and condition, we can generate a training corpus to a feature model based on the character sets. The feature set of character types for constructing a model is the n-gram of characters for backward and forward the word boundaries according to those character types.

5. Experiment and Result

The CRFs algorithm approach will learn the characteristics of text information as a binary classification problem according to a set of type features. Basically, we use an open source software package based on CRFs algorithm, CRF++0.53 (Kudo Taka, 2005-2007) is a simple package, customizable and open source implementation based on the CRFs algorithm. It is able to predict each character from data input and categories it as one of two classes such as: the beginning of a word, and the intra-word characters. Beginning of a word is defined as a labeled class (indicated by “B” in our text corpus), and intra-word characters is defined as a labeled class (indicated by “I” in our text corpus). Based on the machine learning details, we need to generate a text string into two conditions, as shown in Figure 3.

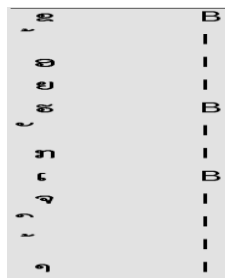


Figure 3: Example of a text generated into character tagged corpus as in word-beginning “B” or intra-word “I” characters

We first purify a text string in where each character is tagged with either word-beginning or intra-word characters. We need to built a tagged corpus as well as possible in CRF format, the tagged corpus in which word boundaries are explicitly marked with special characters, this is a machine learning based that can be verified to analyze a tagged corpus based of type features enclosing these words as boundaries. For the Lao language, we defined the character type for segmentation task into fifteen differences type feature.

We use LaoCORPUS (approximately 100,000 words) to evaluate the performance among different word boundary approaches. We split the text corpus randomly into exam nation and test sets (each set contains 20%). However, we are given a test set of 20% instead, and used the training sets increasing from 20%, 40% and 60% to 80%. The three values of F-measure, precision, and recall are used for performing evaluation.

Value (%)	Size of Text Corpus			
	20K	40K	60K	80K
Precision	75.98	78.43	78.52	80.28
Recall	73.07	76.01	76.77	78.45
F-score	74.49	77.20	77.64	79.36

Table 1: CRFs evaluation by Text corpus size

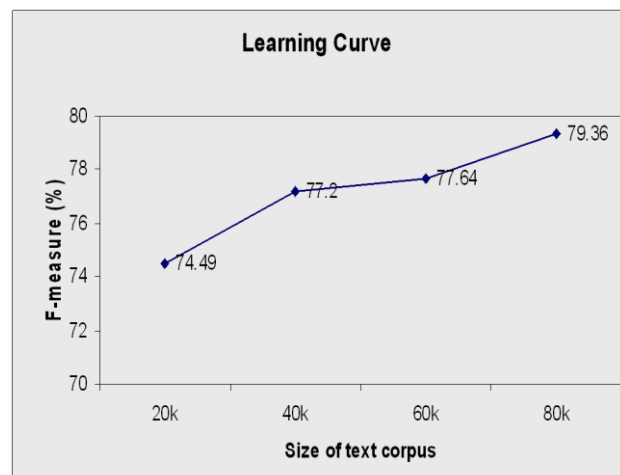


Figure 4: Learning curve from evaluation of text corpus using CRFs algorithm

Otherwise, word segmentation task was compared by our main approach (CRFs model) to another approach such as: dictionary-base (DCB). We also used a test set of 20% instead

from LaoCORPUS the same as previous section, as the result of name entity (NE) that is given details:

NE	Segmented by CRFs	Segmented by DCB
Person	ເພຍ ວິນເຄັນ	ເພຍ ວິນ ເຄ ັ້ນ
Place	ເມືອງ ວໍຊິງຕັນ ດີ ຊີ	ເມືອງ ວໍຊິ ງຕັນ ດີ ຊີ
Company	ບໍລິສັດ ລໍຣິລາດ	ບໍລິສັດ ລໍຣິ ລາດ

Table 2: Comparison of segmented CRFs and DCB

The segmented CRF can be solved these name entities better than the segmented DCB (in Table 2). As the result of CRF, we get a sequence of words correctly as well as using the DCB approach, for example, CRFs can merge |ເຄ|ັ້ນ to be one segment and join it with a previous segment |ວິນເຄັນ| thus the segment has a full correct meaning. The correct answer refers to the segmented word such |ວິນເຄັນ|. To evaluate these approaches are shown below

Approach	Precision	Recall	F1
DCB	80	66.67	72.73
CRFs	80.29	78.45	79.36

Table 3: Evaluation of CRFs and DCB approach

6. Conclusion and Discussion

In this paper, we proposed and compared Dictionary Based and Machine-Learning Based approaches for Lao word segmentation using a tagged corpus. Many previous works have proposed algorithms and models to solve word segmentation problem for languages such as Thai, Chinese and Japanese, however, this research aims to construct a model for Lao language. For the machine-learning based approach, we applied the Conditional Random Fields (CRFs) to train a word segmentation model. We performed the evaluation of a Machine Learning Based approach using CRFs against the Dictionary Based approach. According to the evaluation, the best performance is obtained the CRFs algorithm with

a character tagged corpus. The best result based on the F-measure is equal to 79.36% compared to 72.73% using the dictionary-based approach.

Therefore, to improve the performance further, we need to enlarge the corpus size for training the model. In general, to effectively train a machine learning model especially in NLP tasks, a large size of corpus is needed. For example, compared to Thai word segmentation (Haruechaiyasak *et al.*, 2008), the best performance of F1-measure equal to approximately 96% is achieved with the corpus size of 7 million words. This research will be useful for other applications such as: word line breaking system, machine translation, speech processing (text-to-speech, speech recognition) and image processing.

For future work, we plan to achieve better performance by using syllables (as opposed to characters) as a basic unit for training a model. Another idea is to integrate both the Dictionary Based and Machine-Learning Base approaches, for example, a hybrid approach. The dictionary-base will be used for unknown segment checking on the outputs from the machine-learning base approach.

Acknowledgement

I would like to show sincere gratitude to three people, the first one is my advisor Dr. Choochart Haruechaiyasak who always share me ideas and advices, Second is PAN localization project to support me proceeding the research, Finally, there is my wife who always stays by my side and cheer me up when I confronted with some problem during doing this research.

References

- Choochart Haruechaiyasak et al. 2008. *A Comparative Study on Thai Word Segmentation Approaches*. Proceedings of ECTI Conference, (1) 12-128.
- Choochart Haruechaiyasak and Sarawoot Kongyoung. *LexTo: Thai Lexeme Tokenization*. [Online] March 2006. [Cited 2010 Jan 5]. Available from: <http://www.hlt.nectec.or.th/>
- C. Kruengkrai. and H. Isahara. 2006. *A Conditional Random Field Framework for Thai Morphological Analysis*. Proceedings of the Fifth International

Conference on Language Resources and Evaluation.

Enrique Alba. 2006. *Natural language tagging with genetic algorithms*. Proceedings of Science Direct, (100)173-182.

John Lafferty et al. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of 18th International Conference on Machine Learning, pp. 282-289.

Hanna M. Wallach. 2004. *Conditional Random Fields: An Introduction*. Technical Report of University of Pennsylvania, USA.

Krisda Khankasikam and Nuttanart Muansuwan. 2005. *Thai Word Segmentation a Lexical Semantic Approach*. Proceedings of the 10th Machine Translation Summit, pp. 331, Thailand.

Kudo Taka. *CRF++0.53 yet another CRF Toolkit*. [Online] 2005-2007. [Cited 2009 May 06]. <http://sourceforge.net/projects/crfpp/files>

Ohm Sornil and Paweena Chaiwanarom. 2004. *Combining Prediction by Partial Matching and Logistic Regression for Thai word segmentation*. Proceedings of the 20th International Conference on Computational Linguistics.

Phonepasit Phissamay et al. 2004. *Syllabification of Lao Script for Line Breaking*, Technical Report of STEA, Lao PDR.

Rabiner. L. and Juang. B. 1986. *An introduction to hidden Markov models*. Proceeding of IEEE on ASSP Magazine. (3): 4 – 16.

Sunita Sarawagi. and William W. Cohen. 2005. *Semi-Markov Conditional Random Fields for Information Extraction*. Proceeding of Advances in Neural Information Processing Systems. (17): 1185-1192.

Thanaruk Theeramunkong and Sasiporn Usanavasin. 2001. *Non-Dictionary-Based Thai Word Segmentation Using Decision Trees*. Proceedings of the First International Conference on Human Language Technology Research, Thailand.

Zhao Ziping et al. 2007. *A Maximum Entropy Markov Model for Prediction of Prosodic Phrase Boundaries in Chinese TTS*. Proceeding of IEEE International Conference. pp. 498 – 498.

Burmese Phrase Segmentation

May Thu Win
maythuwin85@gmail.com

Moet Moet Win
moetmoetwin.ucsy@gmail.com

Moh Moh Than
mohmohthanster@gmail.com

Research Programmer, Myanmar Natural Language Processing Lab

Dr.Myint Myint Than
Dr.Khin Aye

Myanmar Computer Federation
 Member of Myanmar Language Commission

Abstract

Phrase segmentation is the process of determination of phrase boundaries in a piece of text. When it comes to machine translation, phrase segmentation should be computerized. This is the first attempt at automatic phrase segmentation in Burmese (Myanmar). This paper aims to express how to segment phrases in a Burmese sentence and how to formulate rules. The system has been tested by developing a phrase segmentation system using CRF++.

1 Introduction

Burmese Language is the national and official language of Myanmar, and is a member of the Tibeto-Burman language family, which is a sub-family of the Sino-Tibetan family of languages. Its written form uses a script that consists of circular and semi-circular letters, adapted from the Mon script, which in turn was developed from a southern Indian script in the 8th century.

Burmese language users normally use space as they see fit, some write with no space at all. There is no fixed rule for phrase segmentation. In this paper, we propose phrase segmentation rules, in linguistics point of view, which will help Natural Language Processing tasks such as Machine Translation, Text Summarization, Text Categorization, Information Extraction and Information Retrieval and so on.

2 Nature of Burmese Language

There are two types of language style - one is literary or written style used in formal, literary works, official publications, radio broadcasts and formal speeches and the other is colloquial or spoken style used in daily communication, both conversation and writing, in literary works, radio and TV broadcasts, weekly and monthly magazines. Literary Burmese is not so much different from colloquial Burmese. Grammar pattern is the same in both, and so is the essential vocabulary. Some particles are used unchanged in both but a

few others are found in one style only. Regional variation is seen in both styles.

2.1 Sentence Construction

One morpheme or a combination of two or more morphemes will give rise to one word; combination of two or more words becomes a phrase; combination of two or more phrases will be one sentence. The following figure shows the hierarchical structure of sentence construction.

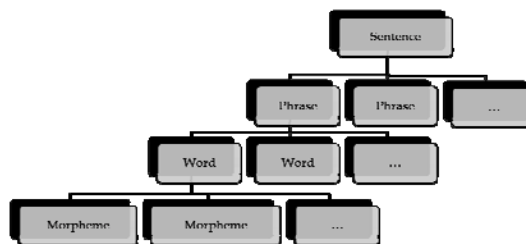


Figure 1: Hierarchical structure of sentence construction

Sentence	သူက ပန်းလေးကို နှမ်းတယ် ။						
Phrase	သူက	ပန်းလေးကို			နှမ်းတယ်		
Word	သူ	က	ပန်းလေး	ကို	နှမ်း	တယ်	
Morpheme	သူ	က	ပန်း	လေး	ကို	နှမ်း	တယ်

Table 1: Sentence construction of a Burmese sentence

In this table, သူက ပန်းလေးကို နှမ်းတယ် means "she kisses the little flower". And သူ is she, က is subject marker, ပန်း is the flower, လေး is little, ကို is object marker, နှမ်း is kisses and တယ် is verb marker.

Morpheme: Morpheme is the smallest syntactic unit that has semantic meaning. The sentence shown in Table.1 has seven morphemes.

Word: The word is the basic item in a sentence.

It consists of one or more morphemes that are linked by close juncture. A word consisting of two or more stems joined together is a compound word. Words that carry meaning are lexical words and words that only show grammatical relation are grammatical words. The sentence shown in Table .1 has six words.

Phrase: Two or more words come together to form a phrase. A phrase is a syntactic unit that forms part of a complete sentence and has its particular place in that sentence. Phrase boundary is characterized by what is called a phrase marker or a pause in speech – open juncture. The phrase marker can be omitted, in which case we say there is a zero marker. Markers show different functions, like subject, verb, object, complement, qualifier, time and place adverb, etc. The sentence shown in Table.1 has three phrases.

Sentence: Finally, we want to say something about the sentence. A sentence is organized with one or more phrases in Subject Object [Complement] Verb or Object Subject Verb order. It is a sequence of phrases capable of standing alone to make an assertion, a question, or a command.

2.2 Syntax

Syntax is the study of the rules and principles found in the construction of sentences in Burmese language. A Burmese sentence is composed of NP+...+NP+VP (where, NP = noun phrase and VP = verb phrase). Noun phrases and verb phrases are marked off by markers but some can be omitted.

3 Parts of Speech

Myanmar Language Commission opines that Burmese has nouns, pronouns, adjectives, verbs, adverbs, postpositions, particles, conjunctions and interjections. In fact, the four really important parts are Nouns, Verbs, Qualifiers or Modifiers and Particles. Pronouns are just nouns. Qualifiers are the equivalents of adjectives and adverbs that are obtained by subordinated use of nouns and verbs. Postpositions and affixes can be considered as markers or particles. Interjections do not count in the parts of speech in Burmese.

4 Phrase Segmentation by Writer's Whim

In Figure.2, sentences are broken into phrases with space in a random way. Phrase segmenta-

tion is employed at the writer's whim; it is not guided by rules.

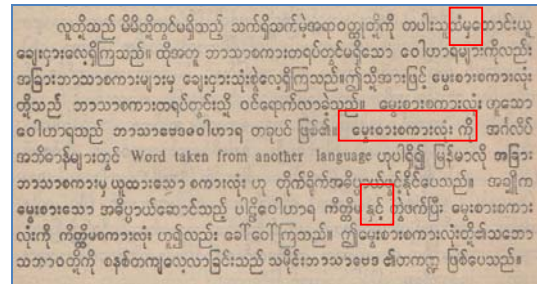


Figure 2: Phrase segmentation by writer's whim
Segmentation may be suggested by pause, or length variety, or clustering of words to bring about meaning. Segmentation in a casual careless way will not be of any help. This paper tries to point out the places where we break sentences with consistency. The boxes shown in the figure are places to break. We will explain how and why we should break at these places in section 6.

5 Particles

We do not normally use lexical words (nouns, verbs and qualifiers) by themselves and they have to be joined by grammatical words (particles or markers) to form a phrase in both literary and colloquial styles. There are three types of particles - formatives, markers and phrase particles.

5.1 Formatives

Formation derives a new word by attaching particles to root morphemes or stems. It may also change the grammatical class of a word by adding affix (prefix or suffix). Adding "စရာ" to the verb "စား eat" gives rise to "စားစရာ food", a noun, and there are many derivational morphemes that change verbs to nouns, verbs to adverbs, and so on.

Reduplication is a word formation process in which some part of a base (a morpheme or two) is repeated, and is found in a wide range of Burmese words. Example; လှိုက်လှိုက် warm → လှိုက်လှိုက်လှိုက်လှိုက် warmly.

It can be obviously seen that formation is a way of word structure. It can be useful sometimes in phrase segmentation, as it can easily be marked off as an independent phrase.

5.2 Markers

A marker or a particle is a grammatical word that indicates the grammatical function of the marked word, phrase, or sentence.

When we break up a sentence we first break it into noun phrases and verb phrases. Verb phrases must be followed by verb markers (sentence-ending or subordinating markers). Noun phrase will be followed by various noun markers, also called postpositions, denoting its syntactic role in the sentence. If we want to show a noun is the subject, a marker that indicates the subject function will be strung with this noun. If we want to indicate a noun to be the object, a marker that indicates the object function will be strung. The distinctive feature of markers is that they show the role of a phrase in the sentence.

noun phrase	noun phrase	noun phrase	verb phrase
ကျောင်းသားများသည်	ပုဂံသို့	လေ့လာရေးခရီး	သွားသည်
The students	to Bagan	an excursion	make

"The students make an excursion to Bagan."

Table 2: A Burmese sentence with markers

In this Table.2, we find three markers,
 - သည် marking the subject of the sentence
 - သို့ marking the place of destination in the sentence
 - သည် marking the verb tense of the sentence

Sometimes, we can construct a noun or verb phrase without adding any visible markers to them. In this case, we say we are using zero markers, symbolized by \emptyset after the noun.

- လေ့လာရေးခရီး suffixing \emptyset marker

We, therefore, use markers as pointers to individual phrases in phrase segmentation of Burmese texts.

5.3 Phrase Particles

Phrase particles are suffixes that can be attached to a phrase in a sentence without having any effect on its role in the sentence. They serve only to add emphasis to particular phrases or to the whole sentence or to indicate the relation of one sentence to another. They are attached to nouns or verbs or even to phrases that already contain markers. Phrase particles are of two types: sentence medial and sentence final.

Example: မင်းကတော့ လာမှာပေါ့နော်။
 You will come, right?

In this example; မင်း: is you (subject), က - subject marker, တော့ - "as for" (sentence medial

phrase particle), လာ - come (verb), မှာ - future (verb marker), ပေါ့ - "of course" (sentence final phrase particle) and နော် - right? (sentence final phrase particle).

6 Markers

Some suffixes mark the role of a phrase in the sentence. Suffixes that perform this function are called "markers". So, markers can be seen as phrase boundaries. Markers can be split into two groups: (1) noun markers and (2) verb markers.

6.1 Noun Markers

Markers that are attached to nouns are called "noun markers". A noun marker shows its function as subject, object or complement, instrument, accompaniment, destination, departure, and others in the sentence. We can sometimes construct a phrase without noun markers. Such a phrase is said to be fixed with zero markers symbolized by \emptyset . Its meaning is the same as that of a phrase with markers. A phrase will be segmented where we consider there is a zero marker in the sentence.

6.1.1 Subject Markers

Marker that marks a noun as a subject of sentence can be defined as subject marker.

Literary Style	Colloquial Style	Translation
သည့် က၊ မှာ \emptyset	က၊ လာ \emptyset	no English equivalent

Table 3: Subject markers and their meaning

Example: (with subject marker)

| ကျွန်တော်က | သေချာမှ လုပ်တယ်။
 | I | like to be sure before I act.

Example: (with zero markers)

| ကျွန်တော် \emptyset | သေချာမှ လုပ်တယ်။
 | I | like to be sure before I act.

6.1.2 Object Markers

Markers that specify the object of the sentence can be defined as object markers.

Literary Style	Colloquial Style	Translation
ကို အား၊ \emptyset	ကို \emptyset	no English equivalent

Table 4: Subject markers and their meaning

Example: သူ | မိန်းကလေးတစ်ဦးကို | ချစ်ဖူးသည်။
He had once fallen in love with | a girl |.

6.1.3 Place Markers

Markers that specify the place and directions can be defined as place markers.

Place Markers	Literary Style	Colloquial Style	Translation
Location	ဤ၊ မှာ၊ တွင်၊ ဝယ်၊ က	မှာ၊ က	at, on, in
Departure	မှ၊ က	က	from
Destination	သို့၊ ကို၊ ဆီ၊ စ...	ကို၊ ဆီ၊ ဆီကို ၊ စ	to
Continuation of place	တိုင်တိုင်၊ အထိ၊ ၊ စ ၊ ...	ထိ၊ အထိ၊ စ	until, till

Table 5: Place markers and their meaning

Example: (Departure)
|နေပြည်တော်မှ| ထွက်လာသည်။
I left | from NayPyiDaw | .

6.1.4 Time Marker

Markers that specify the time can be defined as time markers.

Time Markers	Literary Style	Colloquial Style	Translation
Time	မှ၊ တွင်၊ စ..	မှာ၊ က၊ စ	at, on, in
Continuation of time	တိုင်တိုင်၊ အထိ၊ စ... စ	ထိ၊ အထိ၊ ထက်တိုင်၊ စ ...	up to, till

Table 6: Time markers and their meaning

Example: (Continuation of time)
| ယခုထက်တိုင် | လွမ်းနေဆဲပါ ခိုင်။
I miss you |up to the present, | Khaing.

6.1.5 Instrumentality Markers

Markers that specify how the action takes place or indicate the manner, or the background condition of the action can be defined as instrumentality markers.

Literary Style	Colloquial Style	Translation
ဖြင့်၊ နှင့်	နဲ့	by, with

Table 7: Instrumentality markers and their meaning

Example: | ကားဖြင့် | သွားသည်။
They went | by bus | .

6.1.6 Cause Markers

Markers that specify the reason or cause can be defined as cause markers.

Literary Style	Colloquial Style	Translation
ကြောင့်၊ သဖြင့်၊ ...	ကြောင့်နဲ့	because, because of

Table 8: Cause markers and their meaning
Example: | ဝမ်းရောဂါကြောင့် | သေသည်။
He died | of cholera | .

6.1.7 Possessive Markers

Markers that show a possessive phrase or a modifier phrase can be called possessive markers.

Literary Style	Colloquial Style	Translation
၏၊ ရဲ့၊ ့ (tone mark)	ရဲ့၊ ့ (tone mark)	's

Table 9: Possessive markers and their meaning

Example: | မေမေရဲ့ | ကျေးဇူးကို
အောက်မေ့ပါ သည်။
I remember | mother's | kindness.

6.1.8 Accordance Markers

Markers that specify an action or event occurs in accordance with something can be defined accordance markers.

Literary Style	Colloquial Style	Translation
အလိုက်၊ အရ၊ အလျောက်၊ ...	အရ၊ အတိုင်း၊ အညီ၊ ...	as, according to

Table 10: Accordance markers and their meaning

Example: | ရေစီးအလိုက် | သွားခြင်းကို ရေစုန်ဟု
ခေါ်သည်။
Going | according to the current | is called "downstream".

6.1.9 Accompaniment [coordinate] Markers

Markers that denote accompaniment and two or more items being together with two or more items are accompaniment markers.

Literary Style	Colloquial	Translation

	Style	
နှင့်နှင့်အတူ၊ နှင့်အညီ၊ ...	နဲ့၊ နဲ့အတူ၊ ရော...ရော၊ ...	and, with

Table 11: Coordinate markers and their meaning

Example: မိဘနှင့်အတူ | နေသည်။
She lives together | with her parents |.

6.1.10 Choice Markers

Markers that specify numbers [of persons or things] to make a choice from can be defined as choice markers.

Literary Style	Colloquial Style	Translation
တွင်အနက် အထဲမှ...	တွင်အနက်မှာ၊ ထဲမှ ...	between, among

Table 12: Choice markers and their meaning

Example: | အဖွဲ့ဝင်များထဲမှ | တစ်ယောက်ကို
ခေါင်းဆောင်အဖြစ် ရွေးချယ်သည်။
One person | from among the members |
is chosen as leader of the group.

6.1.11 Purpose Markers

Markers that specify the purpose and are used to denote for, for the sake of, can be defined as purpose markers.

Literary Style	Colloquial Style	Translation
အလို့ငှာ၊ဖို့အတွက်၊ ...	ဖို့အတွက်၊ရန်၊ ...	to, for

Table 13: Purpose markers and their meaning

Example: ကျောင်းသားများသည် | ဗဟုသုတအလို့ငှာ |
လေ့လာရေးခရီးထွက်ကြသည်။
The students set out on a study tour | to
gain experience |.

6.1.12 Demonstratives and interrogatives

Demonstratives and interrogatives may be used in subordination to other nouns, as သည်အိမ်၊
ဟိုအိမ်၊ ဘယ်အိမ် (this, that, which house). They
serve as adjectives followed by nouns. And they
can also be used as independent nouns that can

take noun markers as ဘာကို, ဘယ်မှာ (what,
where). They can be segmented as noun phrases.
Example: | ဘာ | လုပ်ပေးရမလဲ။
| What | can I do for you?

6.2 Verb Markers

Markers that are attached to the verbs are called
"verb markers".

6.2.1 Subordinating Markers

In simple sentences, they are generally at the end
of the sentence and can be seen as independent
markers. We have no need to consider how to
break the sentence into phrases with these mark-
ers because their position plainly shows it. But in
complex sentences, they are in the middle of the
sentence and are known as dependent or subor-
dinating markers. Subordinating markers need to
be considered before breaking a sentence into
phrases. We can break a set of verb and verb
markers attached to it as a verb phrase. Some of
subordinating markers are လျှင် (if), မ---လျှင် (un-
less), ကတည်းက (since), သောကြောင့် (because),
သောအခါ (when) and so on.

6.2.2 Adjectival Markers

Adjectives are formed by attaching adjectival
markers to verbs and they can be segmented as
noun modifier phrases.

Literary Style	Colloquial Style	Translation
သော၊သည့်၊မည့်	တဲ့၊မဲ့	no English equivalent

Table 14: Adjectival markers and their meaning

Example: သူ | ပြောသည့် | စကားကို ကျွန်မ
နားမလည်ချေ။
I didn't understand the words | he spoke |.

6.2.3 Adverbial Marker

Adverbs are formed by adding adverbial marker
“စွာ -ly ” to verbs and they can be segmented as
verb modifier phrases. Adverbs can also be ob-
tained by derivation (prefix and suffix) and re-
duplication of verbs.

Example: (adverbial marker)
| ငြိမ်သက်စွာ | နားထောင်နေကြသည်။
Listen | quietly | .

Example: (reduplication)
| ငြိမ်ငြိမ်သက်သက် | နားထောင်နေကြသည်။

Listen | quietly | .

7 Other Techniques

We can break the sentences into phrases with noun and verb markers. Moreover, we can also segment the following conditions as phrases.

7.1 Complement

A word or a group of word that serve as the subject/object complement can be considered a phrase with zero \emptyset in Burmese.

Example: ဦးညိုမြတ် | သတင်းစာဆရာ \emptyset | ဖြစ်တယ်။
U Nyo Mya is | a journalist | .

7.2 Time Phrase

A word or a group of words that show the time can be defined as a time phrase and can be segmented as a phrase (e.g., မကြာမီ soon).

7.3 Sentence Connector

Grammatical words that are used for linking two or more sentences are called sentence connectors. They are generally placed at the beginning of the second sentence. Some are သို့သော် (but), ဒါကြောင့် (therefore), သို့ရာတွင် (however), ထို့အပြင် (moreover) and so on. We regard them as sentence connectors and break them.

7.4 Interjections

A lexical word or phrase used to express an isolated emotion is called an interjection, for example; အလို (Alas!), အမေ (Oh God) and so on. They are typically placed at the beginning of a sentence. Interjections may be word level or phrase level or sentence level. Whatever level it is, they can be considered a phrase and can be so segmented.

8 Methodology

CRF++ tool is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data. CRF++ will be applied to a variety of NLP tasks.

In our system, we have two phases. The first one is encoding phase and the second one is decoding phase. In encoding phase, at first, we collect and normalize raw text from online and offline journals, newspapers and e-books. When we have sufficient corpus, as preprocessing task, we manually break un-segmented text by the rules mentioned above. Next, we train these sen-

tences decoding with CRF++ tool to get Burmese phrase model. According to our Burmese language nature, we employ unigram template features of CRF implementations.

In decoding phase, un-segmented Burmese sentences are inputted to the system and then automatically encoded with Burmese phrase. As a result, we can achieve Burmese sentences that have been segmented into phrases.

9 Experimental Result

Maximum correctness of phrase segmentation performs when the test and training data come from the same category of corpus. The probability of correctness may be worse if we trained on the data from one category and tested on the data from the other one. Here we tested phrase segmentation of various types of corpus with 5000 and 50000 phrase-model of general corpus respectively.

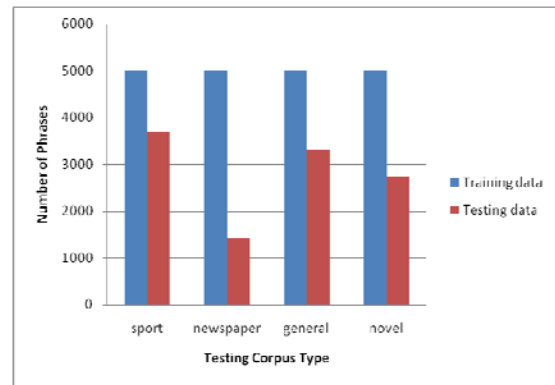


Figure 3: Result of Phrase Segmentation with 5000 phrase-model using CRF++ toolkit

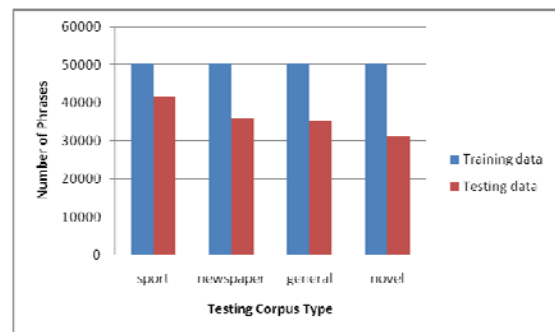


Figure 4: Result of Phrase Segmentation with 50000 phrase-model using CRF++ toolkit

It can be seen that the more sufficient training data, the more efficiency we get. Average scores of phrase segmentation are above 70% according to the F-Measure. The corresponding scores are:

Corpus Type	Score

sport	83%
newspaper	72%
general	70%
novel	62%

Table 15: Various corpus types and their scores

10 Known Issues

Although scores are highly efficient, we face some difficulties that we cannot solve. For example, we can manually segment a sentence into phrases with zero markers such that complement, time and adverbs formed by derivation as a phrase whether it has been attached with markers or not. But in our system, it is difficult to achieve best results because of zero markers. We need more and more training data to cover these zero marker phrases. Boundaries of these phrases may be various. So, we can only get about 50% accuracy for these types of phrases.

Another problem is homonyms. For example: ကို 'Ko' is object marker but it may also be the title of a name like ကိုချမ်းမြေ့ 'Ko Chan Myae'. As a title of a name, we do not need to segment ကိုချမ်းမြေ့ 'Ko Chan Myae'. But CRF++ tool will segment this phrase as ကို 'Ko' and ချမ်းမြေ့ 'Chan Myae' depending on the probability of training data.

11 Conclusion

In this study, we have developed an automatic phrase segmentation system for Burmese language. The segmentation of sentences into phrases is an important task in NLP. So, we have described how we can segment sentences into phrases with noun markers, verb markers, zero markers and other techniques in this paper. We hope this work will help accelerate NLP processing of Burmese language such as Machine Translation, Text summarization, Text Categorization, Information Extraction and Information Retrieval and so on.

12 Further Extension

As we mentioned in section 2.1, the combination of two or more words becomes a phrase. It is easier to segment words after decomposing the phrases of sentence. The result of phrase segmentation will help the word segmentation. Moreover, we can build Burmese parser based on phrase segmentation.

Acknowledgment

The authors are happy to acknowledge their debt and offer grateful thanks Mr. Tha Noe, linguist and Mr. Ngwe Tun, CEO, Solveware Solution.

The authors sincerely want to express acknowledgement to their colleagues of Myanmar Natural Language Processing Team and technicians who helped them and guided on this paper.

References

- J.A. Stewart. 1955. *Manual of Colloquial Burmese*. London.
- J. Lafferty, A. McCallum, and F. Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, In Proc. of ICML, pp.282-289. USA.
- John Okell and Anna Allott. 2001. *Burmese/Myanmar Dictionary of Grammatical Form*. Curzon Press,UK.
- John Okell.1969. *A Reference Grammar of Colloquial Burmese*. London. Oxford University Press.
- Kathleen Forbes.1969.*The Parts of Speech In Burmese and The Burmese Qualifiers*. JBRS, LIII, ii. Arts & Sciences University, Mandalay.
- Pe Maung Tin, U. 1954. *Some Features of the Burmese Language*. Myanmar Book Centre & Book Promotion & Service Ltd. Bangkok, Thailand.
- Willian Cornyn. 1944. *Outline of Burmese Grammar* , Language Dissertation No.38, Supplement to Language, volume 20, No,4.

<http://crfpp.sourceforge.net>

ဦးခင်အေး. မြန်မာသဒ္ဒါနှင့် ဝါ စင်္ဂရစ်ပါးပြဿနာ, အတွဲ (၁၃), အပိုင်း (၅), တက္ကသိုလ်ပညာပဒေသာ စာစောင်.

ဦးဖေမောင်တင်. ၁၉၆၅ အလယ်တန်းမြန်မာသဒ္ဒါစာပေဗိမာန်.

မြန်မာသဒ္ဒါ. ၂၀၀၅.မြန်မာစာအဖွဲ့ဦးစီးဌာန.

Dzongkha Text Corpus

Chungku Chungku, Jurmey Rabgay, Pema Choejey

Department of Information Technology & Telecom

{chungku, jrabay, pchoejey}@dit.gov.bt

Abstract

In this paper, we present the methodology for collecting and compiling text corpus in Dzongkha. The corpus resources is essential for developing applications such as Automatic Part of Speech Detection, Word Segmentation, Text to Speech and Morphological Analysis. This study resulted in building a corpus database containing at least 5 million words collected from relevant websites, print media and manually typed from printed documents. The corpus is tagged with automatic part of speech tagger to enrich the data and to make it more resourceful. In addition, the text corpus is transduced from their original format to an XML format compliant with the XML Corpus Encoding Standard (XCES).

1 Introduction

This is the first attempt ever made to build Dzongkha Text corpus. The objective of this research study is to develop a balanced Dzongkha text corpus which is maximally representative of rich linguistic diversity. This will provide us with huge language resources to be used for developing several natural language processing (henceforth NLP) tools.

A corpus from linguistic point of view is defined as a collection of transcribed speech or written text which have been selected and brought together as a source of data for linguistic research studies. Creation of text corpus from available resources was necessitate due to lack of electronic text in Dzongkha. The corpus was collected from wide range of sources such as the print media, electronic documents and text from websites.

At present the Dzongkha text corpus contains at least 5 million words which are divided into 8 domains and 13 sub domains. First the raw text

undergoes preprocessing, that is the cleaning of data by maintaining its standard format in Unicode. Then it is tokenized into one word per line format using Dzongkha word segmentation¹ tool developed based on lexicon and the longest string matching method.

In order to increase the utility of the corpus, it is annotated using an Automatic Dzongkha part of speech (henceforth called POS) tagger tool (Chungku, et al., 2010) based on Tree tagger (Schmid, 1994). The corpus is automatically annotated with a grammatical tag set containing 66 tags. The corpus is currently being used for other linguistic research studies such as word segmentation, text to speech, morphological analysis and automatic annotation.

Furthermore, to make corpus readily usable by the computers, it is encoded using markup language which marks important language structure including the boundary and POS of each word, sentence, phrase, sections, headings and the meta-textual information. Hence, the text corpus is transduced from the original formats to an XML format with XML corpus encoding standard XCES (Ide, et al., 2000).

Section 2 describes the literature review of the language, section 3 presents the methodology of compilation. Section 4 describes the future work and section 5 concludes the paper.

2 Literature Review

Dzongkha Language

Dzongkha is the national and official language of Bhutan spoken by about 130,000 people in Bhutan. It is a Sino-Tibetan language and is closely related to Tibetan, which was introduced by Thonmi Sambhota. It is an alphabetic

¹ This tool was develop at NECTEC (National Electronics and Computer Technology Center), Thailand.

language with consonants, vowels, phonemes and phonetics characteristics. Like many of the alphabets of India and South East Asia,

Dzongkha Script is syllabic². A syllable can contain one character or as many as six characters. Linguistic words may contain one or more syllables which is separated by superscripted dot called “tsheg” that serves as syllable and phrase delimiters. The sentence is terminated with vertical stroke called “shed”.

Related Works

Initial text corpus, however in lesser size, were used in Dzongkha text-to-speech (Sherpa, et al., 2008), Word Segmentation (Norbu, et al., 2010) and POS Tagger. It was also used for analysis of POS tagset based on Penn Tree Bank. But the corpus was raw in nature, unformatted and unstructured, and had no linguistic annotation.

This study, therefore, expands the initial text corpus base into larger corpus database which is well formatted, structured and linguistically meaningful.

3 Methodology

The process of building text corpus involves two important steps: the planning (design stage) and the execution (creation stage) as described below:

3.1 Design stage

The process of collecting a corpus to its broadest and widest range should be based on its purpose. Our goal was to build corpus that is to be used for multipurpose application both in language technology and general linguistic research. Thus from the initial planning stage certain design principles, selection criteria and classification features was drawn up.

a) Selection Criteria

Three main independent selection criteria, namely domain, medium and time were considered.

Domain

The domain of a text indicates the kind of writing it contains. The data was collected from broad range of domains. The table (1) shows the portable domain percentage:

Domain	Share %
Text Books	25%
Mass Media	50%
World wide Web	10%
Others	15%
Total	100%

Table (1): Portable domain percentage

The corpus contains the written texts from informative writings in the fields of science and medicine, world affairs and political, and leisure. It also contains descriptive writing in the fields of arts, entertainment and finance. More details on domain percentage are found in, cf. table (2).

Medium

The medium of text indicates the kind of publication in which it occurs. The broad classification was made as followed.

- 60% contains periodicals (newspaper etc.)
- 25% contains written text from books
- 10% includes other kinds of miscellaneous published material (brochures, advertising leaflets, manual, etc.)
- 5% includes unpublished written material such as poems, songs, sayings, personal letters, essays and memorandum, etc.

Time

The time criterion refers to the date of publication of the text. Since this is the first attempt made to build Dzongkha text corpus, we collected electronic newspapers published since 2009. This condition for books was relaxed because of its rich linguistic diversity and also due to shortage of electronic books.

b) Classification features

Apart from selection criteria, a few number of classification features were identified for the text in the corpus. Our intention was to make an appropriate level of variation within each criterion, so classification criteria includes following features:

- Sample size or number of words
- Topic or subject of the text

² Omniglot.com. “Tibetan”

<http://omniglot.com/writing/tibetan.htm>

- Authors name
- Text genre and text type
- Source of text

3.2 Creation stage

Creation of Dzongkha text corpus includes following steps:

a) Copyright Permission

Request for copyright permission or clearance was formally sought from the concerned agencies before creation of the text corpus. Prior approval was obtained from the right owners on conditions listed below:

- to allow their materials to be used for creation of text corpus,
- that the text corpus be used for the academic research and not for commercial purpose,

b) Text Collection

Corpus acquisition

Text corpus from different sources and belonging to different domains, as described in table (2), were collected. The acquired corpus contains 5 million tokens (words) approximately.

Domain	Share %	Text Type
1) World Affairs/Political	8 %	Informative
2) Science/Medicine	2 %	Informative
3) Arts/Entertainment	15 %	Descriptive
4) Literature	35 %	Expository
5) Sports/Games	3 %	Procedural
6) Culture/History	7 %	Narrative
7) Finance	10 %	Descriptive
8) Leisure	20 %	Informative

Table (2): Textual domains contained in corpus

Apparently, as seen in the Table (2), the corpus is not well balanced due to lack of electronic text as most of the text are available in printed forms. The highest share (35%) of created corpus belongs to expository text and the lowest share (2%) belongs to informative text in the domain of Science and Medicine.

Corpus Extraction

Extraction of text from the websites was made by studying the richness in the linguistic structure of the text and considering the variety of domain

required. Though there are tools for extraction of texts from websites such as web crawler, we stuck to the process of copying manually from websites (only few websites in Dzongkha is available).

Pre-processing

Cleaning Process: Text cleaning process is divided into two major steps. Firstly, the data gathered was standardized into Unicode character encoding scheme of UTF-8 given its flexibility and portability across platforms. Secondly, correction of spelling mistakes, removal of repetitive and foreign words and deletion of white spaces were performed.

Tokenization

Segmentation of text into words is one of the necessary preprocessing steps for linguistic annotation and also for the frequency analysis of words. POS tagging usually requires a one-token-per line format. This is achieved by using the process of word segmentation. As mention earlier (cf. section 2) Dzongkha belongs to the alphabetic Sino-Tibetan language and is written in continuous form. Therefore, there is no mark to identify word boundary between words.

Word Segmentation: The training data consisting of 40247 token of words was created from existing corpus by segmenting manually to achieve higher accuracy of word boundary. Using this training data and lexicon (dictionary), Dzongkha word segmentation tool (cf. section 1) was developed based on longest matching technique which is a dictionary based method. Then the whole text corpus was tokenized into one word per line based on characters like white space, punctuation marks, symbols etc. The segmentation accuracy of 85.69% is achieved.

c) Encoding of Texts

It is known that for text corpus to be usable by computers, it should be marked-up with its important textual information, in case of Dzongkha text corpus such as:

- The boundary and POS of each word
- Phrase, sections, headings, paragraphs and similar features in written texts
- Meta-textual information about the source or encoding of individual texts

This textual information and others are all encoded by standard mark-up language to help

ensure that the corpus will be usable no matter what the local computational set-up may be.

In addition, texts is further transduced from their original formats to an XML format complaint with XML Corpus Encoding Standard (henceforth XCES) (Ide, et al., 2000). Each corpus file pertaining to different domains is stored in XML structure. The marking is done at word level. This format makes easier for developing web access application for corpus. The design of XCES was strongly influenced by the development of the Guidelines for Encoding of Electronic Text of the international Text Encoding Initiative (TEI).

We found that XML based format is more convenient for corpus since it:

- Supports Unicode
- Programming interface adaptable
- Simplicity, generality and usability mark-up language over internet

The following, cf. (1) shows the example of the how Dzongkha text corpus is encoded using XCES (sentence level).

```
(1) <p id=p1>
    <s id="p1s1">ཀ་ལི་ཕུག་ལུ་འགྲོ་མ་དཀྱི།</s>
    <s id="p1s2">ཁ་ལུ་ཟ་ནི་མིན་འདུག།</s>
    <s id="p1s3">ག་ཉི་མལ་ལ་ཉི་འགྲོ་ཅུང་།</s>
</p>
```

d) Linguistic Annotation of Text (“POS tagging”)

POS tagging means annotating each words with their respective POS label according to its definition and context. It provides significant information for linguistic researcher with morphological, syntactic or semantic kind. Such enriched data is useful, especially when designing higher level NLP tools.

In this research study, the morpho-syntactic annotation is automatically added to the entire corpus using a probabilistic tagger developed (Chungku, et al., 2010). Annotated texts produced by this automatic tagger uses tag set³ containing 66 tags, its design is based on Penn Guidelines⁴ (though there is some changes made to fit the language structure). The highest

³ The original Dzongkha tag set is described at <http://www.panl10.net>

accuracy achieved by this automatic tagger is about 93.1%. The accuracy is further enhanced by performing manual post-edit thereby resulting in better annotated texts. Table (3) shows an example of how the process of automatic tagger takes place.

Input text	Output text	
Word	Word	POS tag
འབྲུག་	འབྲུག་	NNP
གི་	གི་	CG
རང་ལུགས་	རང་ལུགས་	NNP
འཆམ་	འཆམ་	NN
།	།	PUN

Table (3) Example of automatic annotation

e) Storage, Documentation and Web Interface

In the last stage, we manually added detailed descriptive information to each text in the form of header. The header information contains specific information such as author’s name, source of the text, etc. which is useful for computer programming.

A web based interface to provide easy access to the corpus is being developed. The corpus database thus built can be made available on CD ROM for research purposes.

4 Future Work

This is the first attempt ever made to build the corpus database. Therefore, there are enough rooms for improvements in terms of quality and usefulness.

Increasing the text corpus size may lead to further improvement in tagging and segmentation accuracies thereby leading to better quality annotated text corpus.

Tools for collection of text corpus may be explored. Optical character recognition (OCR) system being currently developed by the department may ease the collection process.

In addition, balancing corpus from broad ranges of domains and annotating text using other annotation techniques may improve the quality.

⁴ The Penn Guidelines can be downloaded from: <http://www.cis.upenn.edu>

5 Conclusion

Corpus is very important and is the basic resource for language processing and analysis. This document demonstrates the collection and compilation methodology of Dzongkha text corpus. It also demonstrates how the corpus is automatically annotated with automatic POS tagger. The corpus database contains 5 million tokens of words. The corpus is being used for Dzongkha word segmentation, automatic corpus tagger and advanced text-to-speech system.

Furthermore, it is expected that the corpus will become extremely useful for developing other language processing tools such as lexicon, machine translation and frequency analysis.

Acknowledgments

This research work was carried out as part of the PAN Localization Project (www.PANL10n.net) with the aid of grant from International Development Research Center (IDRC), Ottawa, Canada, administered through the Center of Research in Urdu Language Processing (CRU LP), National University of Computer and Emerging Sciences (NUCES), Pakistan. The research team would also like to thank media agencies (Bhutan Observer, Bhutan Broad Casting Service, Druk Nyeltshul, Bhutan Today) in Bhutan, Bhutan National library and Dzongkha Development commission for their valuable contribution that made this research successful.

References

- British National Corpus. 2009. *Reference Guide for the British National Corpus (XML Edition)*. Retrieved October 30, 2009, from <http://www.natcorp.ox.ac.uk/>
- Chungku, Chungku, Gertrud Faaß and Jurmey Rabgay. 2010. *Building NLP resources for Dzongkha: A Tag set and a tagged Corpus*. Proceedings of the Eighth Workshop of Asian Language Resources (WS1), 23rd International Conference on Computational Linguistics (COLING 2010), 103-110, Beijing, China,
- Nancy Ide, Laurent Romary, Patrice Bonhomme. 2000. *XCES: An XML-based standard for Linguistic Corpora*. In proceedings of the Second Annual conference on language Resources and Evaluation, , 825-30. Athens.
- Nancy Ide, Randi Reppen, Keith Suerman. 2002. *The American National Corpus: More Than the Web can provide*. Retrieved November 1, 2010, from <http://www.cs.vassar.edu>.
- Sithar Norbu, Pema Choejey, Tenzin Dendup, Sarmad Hussain, Ahmed Mauz. 2010. *Dzongkha Word Segmentation*. Proceedings of the Eighth Workshop of Asian Language Resources (WS1), 23rd International Conference on Computational Linguistics (COLING 2010), 95-102, Beijing, China.
- Asif Iqbal Sarkar, Shahriar Hossain Pavel, and Mumit Khan. 2007. *Automatic Bangla Corpus Creation*. PAN Localization Working Papers, pages 22-26, 2004-2007.
- Helmut Schmid. 1994. *Probabilistic Part-of-Speech Tagging Using Decision Tree*. Proceedings of the International Conference on New Methods in Language Processing, pages 44-49, Manchester, UK.
- Eden Sherpa, Dawa Pemo, Dechen Choden, Anocha Rugchatjaroen, Ausdang Thangthai, Chai Wutiwatchai. 2008. *Pioneering Dzongkha text-to-speech Synthesis*. Proceedings of Oriental COCOSDA, pages 150-154, Kyoto, Japan.
- Martin Wynne (Ed.). 2005. *Developing Linguistic Corpora: a Guide to Good Practice*. Oxford: Oxbow books.
- XML Corpus Encoding Standard Document XCES 1.0.4. 2008. *XCES Corpus Encoding Standard for XML*. Retrieved August 27, 2009, from <http://www.xces.org/>

Towards a Sinhala Wordnet

Viraj Welgama, Dulip Lakmal Herath, Chamila Liyanage, Namal Udalamatta, Ruvan Weerasinghe

Language Technology Research Laboratory,
University of Colombo School of Computing, Sri Lanka

{vww, dlh, cml, ugn, arw}@ucsc.lk

Tissa Jayawardana
Department of Linguistics,
University of Kelaniya,
Sri Lanka

Abstract

This paper describes the methods adopted and the issues addressed in building a Sinhala Wordnet, based on the Princeton English WordNet (PWN). Its aim is to develop the most important parts of a wordnet for the Sinhala language, in order for it to be of optimal use without being complete. The importance of entries were estimated using the word frequencies of the 10 million word UCSC Sinhala corpus of Contemporary Sinhala, and the relevant lexico-semantic relations extracted from the PWN. The paper describes how the Sinhala Wordnet was developed with a view to presenting a recommended strategy for other languages for which wordnets may be developed in the future.

1 Introduction

Wordnet is one of the most useful lexical resources for many key natural language processing and computational linguistic tasks including Word Sense Disambiguation, Information Retrieval and Extraction, Machine Translation, and Question Answering among others. It is based on the theories developed in Lexical Semantics and defines different senses associated with the meaning of a word and other well-defined lexical relations such as synonym, antonym, hypernym, hyponym, meronym and holonym.

The Princeton WordNet (PWN) (Fellbaum, 1998), is a large lexical resource developed for English, which contains open class words namely; nouns, verbs, adjectives and adverbs. These words have been grouped together based on their

meanings, with a single set of such synonyms being called a *synset*. Many efforts have been reported in recent years to develop such lexical resources for other languages (e.g. Darma Putra et. al. (2010), Elkateb et al, (2006) among others) based on the relations defined in the PWN.

Sinhala is an Indo-Aryan language spoken by a majority of Sri Lankans. It is also one of the official and national languages of Sri Lanka. The University of Colombo School of Computing (UCSC) has been involved in building Sinhala language resources for NLP applications for many years. Some of these include a 10 million word Sinhala corpus, a part-of-speech tag set, and a tri-lingual dictionary. The motivation behind the project to build a Sinhala wordnet is to fulfill the requirement of a semantico-lexical resource for NLP applications.

A brief overview of three prominent wordnet projects namely the PWN, the *Euro WordNet* (Vossen, 2002), and the *Hindi WordNet* (Narayan et. al. (2002) and Chakrabarti and Bhattacharyya (2004)) were closely examined as a part of the Sinhala wordnet development project, to understand the approaches taken, structures used, language specific issues and the functionalities available in them. Using this input, it was decided to define the relations among Sinhala words using PWN sense IDs in order to keep the consistency with many other wordnet initiatives in the interest of possible interoperability. This also helped in developing the Sinhala wordnet with less effort, by using the linguistic notions that held across languages and language families.

The initial idea that the PWN synsets could be directly translated for use as the Sinhala Wordnet had to be abandoned owing to the top-level categories in it being less relevant in tasks such as word-sense disambiguation owing to the lack of

ambiguity in them in the Sinhala language. Instead, the UCSC Sinhala corpus, which consists of 10 million Sinhala words in contemporary use, was used as the main resource to base the selection of the most important parts of the wordnet which needs to be built to be of use for applications for Sinhala. High frequency open class words from the corpus were identified in order to discover word senses that contributed most to contemporary language use. Each of these words was then considered as a candidate for inclusion in the Sinhala wordnet sub-set to be constructed first. Other senses relating to these words were then enumerated in consultation with language and linguistics scholars. This strategy helped to build the Sinhala wordnet in a phased manner, starting with most prominent and hence multi-sense words in the language.

This paper presents the work carried to develop the Sinhala wordnet using the PWN synset IDs. The rest of paper will describe the methodology, challenges and the future work of the Sinhala wordnet project.

2 Methodology

A survey of potential resources for the Sinhala wordnet project was carried out at the beginning of the project. As a result of this survey, it was found that the tradition of thesaurus building is not new to Sinhala language studies but has been in general fairly well established in traditional linguistic studies originating from ancient India.

Though there are some Sinhala language resources available in the Sinhala literature which are closer to the current work, many of these could not be directly used due to poor coverage of contemporary Sinhala (mainly covers traditional ancient language) and the poverty of concept classification (confined to religious and preliminary concepts). Having examined them thoroughly one main resource and a couple of supplementary resources were identified as primary sources for the project. A few popular Sinhala dictionaries and thesauri were among these (e.g. Wijayathunga, 2003).

The literature concerning the semantic aspect of the Sinhala language is relatively limited due to it not being handled formally by scholars of Sinhala language research. This has led to a situation where it is difficult to express the semantics of words and their sense relations accurately. In order to address these issues, it was decided to complement the information given in such Sinhala language resources in an informal manner by

working with linguistic scholars who have a strong theoretical background in both traditional grammar and modern linguistic analysis of Sinhala and English languages.

Having closely studied the approaches taken in other wordnet initiatives, a strategy for the development of the Sinhala wordnet was established. Many wordnet initiatives have used a top-down approach, in which abstract concepts have been enumerated starting with a kind of upper level ontology and then gradually working down over many decades. Owing to time and resource limitations, we had to use a more data-driven approach to clearly identify the most important subset of senses within a wordnet that would be of most value to researchers. As a significant quantum of work has been done in the PWN in terms of building the infrastructure for all later wordnets, our strategy was developed in such a way that lessons learnt from the PWN project could be used to avoid most of the hurdles that have been negotiated by the developers of the PWN.

Figure 1 shows the workflow of the development process of the Sinhala Wordnet. The steps of the methodology of the Sinhala Wordnet project can be divided into sub tasks as described below.

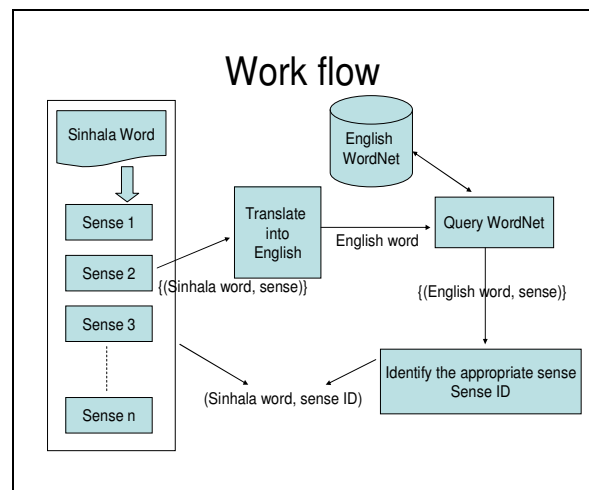


Figure 1. Workflow of Sinhala Wordnet Development

2.1 Word Selection Process

At the outset, the words to be considered for inclusion in the Wordnet were chosen from the UCSC Sinhala Corpus according to their frequency. The most frequently occurring 500 words excluding function words were chosen to

build the prototype of the Sinhala Wordnet. Next this was expanded to include the top 1000 words once the strategy was well established. As Sinhala is a morphologically rich language there are many different word forms for a given base form and only one single form called *lemma* is selected for the current system. In cases where a word form different to the base form had a different semantic value, that form was considered as a separate entry. Some words have alternate spellings and phonological variations that have led to semantic variations and such words are also considered as separate entries in wordnet.

2.2 Sense Identification Process

As discussed in Section 2.1, one word can have more than one sense and it is extremely difficult to identify all the senses of a given word. We followed two approaches to identify the senses of words, namely dictionary look up and look up of English translations of the corresponding word in the PWN. Finally, a linguistic scholar determined the list of senses for a given word after reviewing the potential senses given in the dictionary and the PWN. The main source for extracting Sinhala word senses was *Maha Sinhala Sabdakoshaya* (Wijayathunga, 2003), which is the major Dictionary of the contemporary Sinhala language.

2.3 Sense Relation Extraction

PWN defines six main word sense relations, namely, synonyms, antonyms, hypernyms, hyponyms, meronyms and holonyms. As defining them from scratch is time consuming and requires a sophisticated expertise in lexical semantics, it was decided to extract them from the PWN database and store them in a human readable format. The main motivation behind this decision was the fact that a majority of the senses are language and culture independent. Therefore this approach helps incorporating Sinhala words with relations given for English, in order to build the Sinhala wordnet with less effort.

2.4 Sinhala to English Translation and PWN Query

The accurate English translation for a given sense of a Sinhala word was determined by a linguistic scholar conversant in both Sinhala and English language usage. Having precisely translated the Sinhala word sense into English, it is in turn looked up in the PWN to obtain the relevant *synset identifier*.

2.5 Sense ID Assignment

The Sinhala word with a particular word sense is then inserted to the Sinhala wordnet database with the sense identifier obtained according to the step described in 2.4. This process helped to maintain all the sense relations, which have already been defined in the PWN database, automatically and with no extra effort on our part.

2.6 Gloss Translation

After identifying the exact sense ID for a given word-sense, we used the knowledge of expert translators to translate the gloss defined in the relevant PWN entry into Sinhala. Translators were given the freedom to change the gloss according to the language-culture of Sri Lankan Sinhala, when the PWN gloss was found to be not appropriate for the context.

2.7 Synset Identification

When the sense ID, POS and the gloss was determined for a given sense, native speakers knowledge and the other resources such as dictionaries and thesauri were used to identify the corresponding synset for that sense. This was manually done by two language experts.

The senses identified through above process were stored in an Excel sheet (Figure 2) and currently has not been integrated with any user interface. More details on data storage are explained under Future Work.

Synset	PWN ID	POS	Gloss
සදහන/සටහන/විස්තරය	06418196	n	කෙටි ලිපික සටහන
ඉන්නවා	02703136	v	නැතහොත් තනිවයන නැවතී සිටිනවා
මුලික	01922424	s	මුලික දත්ත හෝ ප්‍රතිපත්තිලිපි අඩුම වන්නා වූ
ඒවිකය	13777175	n	ඒවිකවේදන පොතක සාකච්ඡා හෝ රටාව
වැඩ/රාජකාරිය/කාර්යය/කටයුතු	00570312	n	යම්කිසිවක් සඳහා හෝ සෑදීම කෙරෙහි යොමු වූ ක්‍රියා
කෙරුණ	08373013	n	මෙම ඒවික වන රට, ප්‍රාන්තය හෝ නගරය
අදහස/බිතුම්/ලැබීමකය	05761049	n	සංකල්පනාව, ප්‍රජානනයේ දැනටත්වනය
පමණ/තරම/මනෝරාමියාව/පස	00032028	n	ප්‍රමාණ කළහැකි යම්කිසි දෑහි ප්‍රමාණය
සුරතලා	00479055	v	අවසාන කරනවා හෝ අවසානයට පැමිණෙනවා
විනයය	07233906	n	යම්කිසි පමණකට හෝ සිදු කිරීමක සිදුවිය (හෝ සි)
මනාලිය/මනාලිය	05841869	n	යම් විශ්වාසයක් තබනු ලැබූ අනුමාන අදහසක්
මුලික	01051890	s	සාර්ථකයක් ලෙස හෝ සඳහාමක් ලෙස වෙහෙස කරා
නිර්මාණය කරනවා	01607166	v	කෘතීම නිෂ්පාදනයක් ඇති කරනවා
සාමාන්‍යයෙන්	00491749	r	ස්වාභාවික හෝ සාමාන්‍යය දැක්වීමට
නැවත/සාප්ත/සාධිත/සාධිත/සාධිත	00041086	r	සාමාන්‍ය, අන්තිම
ජාතිකයා	03070805	a	ජාතියකට හෝ රටකට අයත්
දෙනවැස්ස/සිළුදෙන	01868513	n	දෙනුව, විශේෂයෙන් ක්ෂීරපායී සතුන්ගේ ස්ත්‍රී ලිංග

Figure 2. Sinhala Wordnet Database

3 Challenges for ‘new’ languages

Several linguistic issues need to be addressed in order to capture language specific features in the design of the system. Most of these occurred owing to the morphologically rich nature of the Sinhala language, as well as the cultural biases of the English Wordnet as used in the PWN. The major needing resolution in the development process can be categorized as follows:

3.1 Morphological Forms

As mentioned above, Sinhala is a morphologically rich language which accounts for up to 110 noun word forms and up to 282 verb word forms. Therefore it is extremely important to incorporate a morphological parser to map such word forms to their corresponding lemmas. Table 1 shows some examples these morphological forms for moth nouns and verbs. A complete morphological parser for Sinhala is being developed at the Language Technology Research Laboratory (LTRL) of the UCSC and is expected to couple with the Sinhala wordnet to enhance the value of this resource.

Morph. Form	POS	Meaning	Lemma
බලමි (<i>balāmi</i>)	Verb	See (1 st person, Sg)	බලනවා (<i>balānāvā</i>)
බැලිය (<i>bælīyā</i>)	Verb	See (3 rd Person, Sg)	බලනවා (<i>balānāvā</i>)
බලද්දී (<i>baladdī</i>)	Verb	While Seeing	බලනවා (<i>balānāvā</i>)
බල්ලෝ (<i>ballō</i>)	Noun	Dog (Nominative, Pl)	බල්ලා (<i>ballā</i>)
බල්ලන් (<i>ballan</i>)	Noun	Dog (Accusative, Pl)	බල්ලා (<i>ballā</i>)
බල්ලාගේ (<i>ballāgē</i>)	Noun	of Dog	බල්ලා (<i>ballā</i>)

Table 1. Morphologically different forms which share the same lemma

3.2 Compound Nouns and Verbs

Compounding is a very productive morphological process in Sinhala. Both Sinhala nouns and verbs formed by compounding nouns (nouns) and nouns with verbs (e.g. verbs *do* and *be*) are extremely productive. As a result of this compounding, the original sense of the constituents of the compound noun is altered, resulting in the derivation of a new sense. The methodology we used to extract the most important senses (as explained in 2.1) does not detect compound words, since we used the most frequent *single words* extracted from the corpus.

3.3 Language and Culture Specific Senses

Several culture specific senses were among the most frequent Sinhala words which had no corresponding sense IDs defined in PWN (e.g., “මිරිස් ගල” *miris galā* - “A flat stone and drum stone use to grind chilly, curry powder etc.”,

“පොළ ගනවා” *pol gānāvā* - “The act of scraping coconuts using a coconut scraper”). Two possible approaches were identified to find the appropriate place in the ontology for such senses. The first was to find the closest approximation in the existing ontology for an equivalent concept. The second was to extend the ontology appropriately to accommodate these concepts in order to represent them most accurately.

3.4 Word Selection Criteria

The words for the Sinhala wordnet were chosen from the UCSC Sinhala Corpus as described in Section 2.1. Many of these words have senses in standard Sinhala dictionaries that are not used in contemporary Sinhala. It was identified that taking these senses of words into account is not useful for the goals of the current project, and therefore they were ignored after carefully examining the period to which the usage of such senses belong.

4 Future Work

The process of building a Sinhala wordnet was mainly targeted as a resource for aiding language processing tasks. Hence aspects of providing an integrated GUI were not given priority and the resource stands on its own as a structured text document. It is expected to be integrated with a Sinhala morphological parser (which is currently being developed) in order to be of practical use. Therefore it is necessary to integrate this lexical resource with a comprehensive tool for manipulating data easily.

The current Sinhala Wordnet consists of 1,000 of the most common senses of contemporary Sinhala usage. Lexical relations of these words have been automatically linked to the English Wordnet due to adopting PWN sense IDs, even though some entities related to these 1,000 words are not present in English. Therefore it is essential to expand the Sinhala wordnet for these links and also to add senses according to importance, in order to build a comprehensive Sinhala lexical resource.

The AsianWordNet (AWN) Project of the TCLLab of NECTEC in Thailand is an initiative to interconnect wordnets of Asian languages to which the present Sinhala Wordnet is being linked. It is hoped that this effort will lead to a comprehensive multi-lingual language resource for Asian languages.

5 Conclusion

Building a lexical resource such as wordnet is essential for language processing applications for the less resourced languages of the world. However the task requires significant resource allocations and expert knowledge to build for a particular language. As such, if a 'newly digitized' language can benefit from already developed linguistic infrastructure for another language, much effort can be saved. In the process of such adoption however, certain adaptations may need to be performed owing linguistic and cultural peculiarities of the language concerned.

This paper recommends the use of corpus statistics to identify the most important senses for a particular language to encode in a wordnet, in any given phased implementation effort. Such statistics provide a way to identify the most frequently used word senses specific to a culture which need to be dealt with first in order to get the highest return on investment of effort.

For languages which are morphologically rich, a morphological parser needs to be incorporated as a front end to such lexical resources. Many of the most frequent words of this kind of agglutinative language are irregular in form, requiring a morphological analyzer able to handle such forms.

Acknowledgements

This work was carried out under Phase 2 of the PAN localization project funded by IDRC of Canada. Authors acknowledge the contribution of several members of the Language Technology Research Laboratory of the University of Colombo of School of Computing in building the Sinhala Wordnet. In particular, the contribution of Vincent Halahakone in proofing Sinhala-English translations using his immense experiences as an English language teacher for government schools and universities is gratefully acknowledged. The authors also acknowledge the feedback given by two reviewers of this paper which helped in improving the quality of the work reported. Any remaining errors however, are those of the authors.

References

Chakrabarti, D. and Bhattacharyya, P. 2004, *Creation of English and Hindi Verb Hierarchies and their Application to Hindi WordNet Building and English-Hindi MT*, Global WordNet Conference (GWC-2004), Czech Republic.

Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande and P. Bhattacharyya 2002, *An Experience in Building the Indo WordNet - a WordNet for Hindi*, First International Conference on Global WordNet, Mysore, India.

Elkateb, S., Black, W., Rodríguez, H., Alkhalifa, M., Vossen, P., Pease, P., Fellbaum, C. 2006. *Building a WordNet for Arabic*. LREC, Italy.

Fellbaum, C. (ed) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Putra, D.D, Arfan, A., Manurung R. 2010. *Building an Indonesian WordNet*, University Indonesia.

Roget, P.M , Roget, J. L. , Roget, S. R. 1962. *Thesaurus of English Words and Phrases*. Penguin Books.

Vossen, P. (ed) 2002. *Euro WordNet General Document*. Vrije Universiteit, Amsterdam

Wijayathunga, Harischandra 2003. *Maha Sinhala Sabdakoshaya*. M. D. Gunasena & Co. Ltd., Colombo

CorpusCollie

A Web Corpus Mining Tool for Resource-Scarce Languages

Doris Hoogeveen^{1,2}
¹_textkernel
Amsterdam, The Netherlands
dl_doris@hotmail.com

Guy De Pauw²
²CLiPS - Computational Linguistics Group
University of Antwerp
Antwerp, Belgium
guy.depauw@ua.ac.be

Abstract

This paper describes CORPUSCOLLIE, an open-source software package that is geared towards the collection of clean web corpora of resource-scarce languages. CORPUSCOLLIE uses a wide range of information sources to find, classify and clean documents for a given target language. One of the most powerful components in CORPUSCOLLIE is a maximum-entropy based language identification module that is able to classify documents for over five hundred different languages with state-of-the-art accuracy. As a proof-of-concept, we describe and evaluate the fully automatic compilation of a web corpus for the Nilotic language of Luo (Dholuo) using CORPUSCOLLIE.

1 Introduction

In the field of human language technology, corpora are the cornerstone to the development of robust language technology tools and applications, such as part-of-speech taggers, syntactic parsers, machine translation and text-mining systems. Many of these systems use some kind of statistical processing and the adage “*There is no data like more data*” has never been more relevant: many studies suggest that a system’s accuracy is often a function of corpus size. This unfortunately also means that for *resource-scarce* languages, the full language technological potential can often not be released.

Before the Internet era, corpora for resource-scarce languages were almost impossible to get hold of, unless one went through the slow and tedious

process of collecting and digitizing (often severely outdated) printed works. Not until recently have researchers started looking at the Internet as a valuable source for language data and corpus material (Baroni and Bernardini, 2006). *Web mining* corpora has proved to be a relatively fast and cheap way to harvest digital language data. A lot of web mining approaches have been described over the years (Resnik, 1999; Ghani and Jones, 2000; Scannell, 2007; Kilgarriff et al., 2010) and quite a few of those tools are publicly available (Baroni and Bernardini, 2004; CorpusCatcher, 2011).

The CORPUSCOLLIE web mining tool described in this paper, differs from the current state-of-the-art in that it attempts to incorporate a wide range of text processing and classification modules that ensure the resulting corpus is as clean and expansive as possible. It furthermore adheres and makes use of the authoritative Ethnologue classification system. In this paper we will outline the different components of CORPUSCOLLIE and will provide a quantitative evaluation of the tool as a web miner for the resource-scarce, Nilotic language of Luo (Dholuo).

2 Harvesting data

As a very first step, the user is asked to specify the ISO 639-3 language code, as defined by Paul (2009), as well as a number of user-definable parameters (see below). The user will also need to obtain a Bing API AppID, which is free of charge¹. Unfortunately other major search engines such as Google and Yahoo do no longer provide such APIs.

¹<http://www.bing.com/developers>

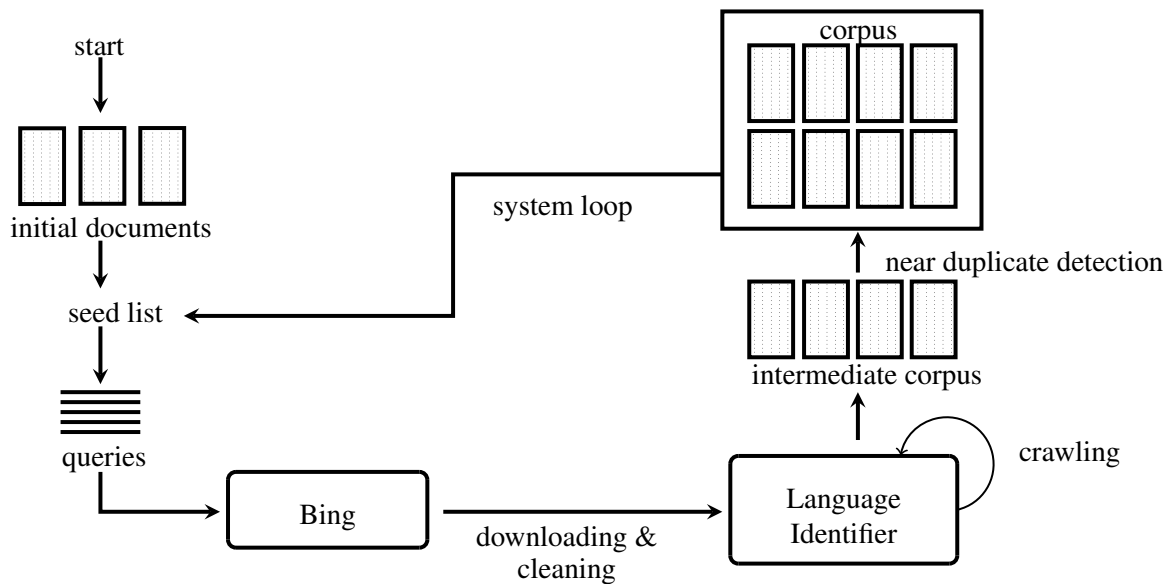


Figure 1: General Architecture of CORPUSCOLLIE.

The general architecture of CORPUSCOLLIE is outlined in Figure 1. The tool starts out with a number of documents in the target language from which a *seed list* of words is extracted. Words from the seed lists are randomly combined into search queries, which are sent to the search engine. Candidate documents in the target language are then classified by a language identification module and added to the final web mined corpus, provided they pass the (near) duplicate detection filter. From the newly found documents, a new seed list is generated and the whole process starts anew. In this section, we will go into more detail on the respective components of the CORPUSCOLLIE tool.

2.1 Query generation

The default settings of CORPUSCOLLIE assume that the user has a number of documents available in the target language. These need to be in text-only format and should be encoded in UTF-8. From these documents, a seed list is generated, containing the n most frequent words in the initial documents².

Highly frequent words are often short function words however, and many frequently occurring ho-

²It is also possible to skip automatic seed list generation and start the web mining process with a manually defined seed list.

mographs are shared among languages³. Obviously, such words do not constitute appropriate search terms to mine one particular target language. We therefore downloaded all of the available Wikipedia data. For those languages for which a critical amount of Wikipedia data was available (± 240), we constructed a *safetynet*: if a word in the seed list occurs in one of the languages of the safetynet file, it is not retained for possible search query generation. This safetynet helps ensure that the seed list contains the most frequent words of a language, that are not common in other languages.

From this seed list, a number of search queries are generated by randomly combining three words from the seed list. These queries are then sent to the Bing Search API module⁴, which returns a list of URLs that match the search query in question.

We tried to reduce the risk of getting pages in the wrong language even further by specifying the domain extensions the search engine needed to take into account. For this purpose we used Ethnologue data to construct a database where each language code is listed with the countries where it is spo-

³For example, “*san*” is among the 100 most frequent words in more than 30 languages.

⁴<http://uswaretech.com/blog/2009/06/bing-python-api>

ken, and the associated Internet domain extensions⁵. These extensions were appended to the search engine queries, together with more general domain names (e.g. .com, .biz, .org, ...). It seems however, that Bing does not yet handle extensive lists of domain specifications well and this functionality is currently disabled by default.

2.2 Processing the documents

CORPUSCOLLIE only harvests HTML pages from the Internet, as automatic conversion of PDF files and other legacy formats is unreliable, often leading to messy results and noise in the web mined corpus. Once it has been established that the URL returned by Bing, points to a true HTML page, we download it. Most web miners use external tools with crawling functionality for this task. We have opted for a standard python module (URLLIB) to download the pages. This allows us to save bandwidth by downloading a page from a web site first and then decide whether or not to crawl the rest of the site, based on the decision of the language identification module.

Encoding - A very important, but all too often ignored issue is that of encoding. We want the web mined corpus to be uniformly encoded in UTF-8. CORPUSCOLLIE uses the Python version of the Universal Encoding Detector, CHARDET⁶, which displays the encoding of a page with a reliability score. If the encoding of a web page is determined with a reliability of over 70% (manually defined threshold), we keep the page, after converting it to UTF-8 when necessary.

HTML to (clean) text - After downloading the pages, they need to be *cleaned*. HTML needs to be converted to plain text, JavaScript, css and other forms of code need to be removed, as well as comments. We opted to use the Python module HTML2TEXT⁷ to convert HTML pages to *Markdown* format, a plain text format that attempts to preserve the structure of the text, by using simple, non-obtrusive markings such as square brackets around links, and hash signs to indicate headers.

Boilerplate is a term denoting all the linguistically uninteresting parts of web pages, such as navigation bars and disclaimers. These need to be re-

moved. Unfortunately, there are no unambiguous cues that indicate if something is part of the boilerplate of a page or not. Several researchers (Baroni and Bernardini, 2006; Sharoff, 2006; Ferraresi et al., 2008) have used a reimplementations of the BTE tool (Finn et al., 2001), which uses tag density as a heuristic. Other suggested techniques for boilerplate removal work on the basis of document size (Fletcher, 2004; Kilgarriff et al., 2010), word count (Ide et al., 2002) or the presence of function words (Ferraresi et al., 2008). Unfortunately, these approaches are too slow to be used as a module in our tool or are not publicly available.

The user can ask CORPUSCOLLIE not to remove Boilerplate at all (leaving open the option of using the aforementioned techniques post-hoc). Alternatively, a module can be enabled that uses regular expressions to remove patterns that are likely to be boilerplate. For instance, a sentence with a ©sign in it, is likely to be a disclaimer and can be removed. A pattern of words with pipe-signs (|) is likely to be a list of links at the end of a page. At this time, we have no experimental results on the effectiveness of our boilerplate removal algorithm.

2.3 Filtering the results

The cleaned pages are then classified two times. An extensive language identification module (see Section 3) checks whether the document is indeed written in the target language. If that is the case, CORPUSCOLLIE can be configured to crawl the rest of the site up to a user-defined depth, as it is likely that the same web site has more pages in the target language.

Finally, each page is classified by a (near) duplicate detection module. Near-duplicates are documents that share a significant portion of their content with a page already present in the corpus. These need to be removed because having identical documents in the corpus will negatively affect statistical processing.

For an extensive overview of the existing work on near duplicate detection, we would like to refer to Kumar and Govindarajulu (2009). CORPUSCOLLIE makes use of the *simhash* method, described in Charikar (2002), because of its speed and its ability to accurately perform near duplicate detection on smaller documents.

⁵For example Luo is associated with .ke, .tz and .ug.

⁶<http://chardet.feedparser.org>

⁷<http://www.aaronsw.com/2002/html2text.py>

2.4 Looping

Figure 1 shows the *system loop* link. Once we have harvested more data, we can construct a new seed list and repeat the entire mining operation again. This iteration can in principle be performed ad infinitum, but CORPUSCOLLIE is configured to loop a user-defined number of times.

The way the seed list is constructed in the second (and subsequent) iteration(s), differs from the very first iteration. Ghani et al. (2001) identified the *odds ratio* of a word as the best metric to select candidate seed words. The odds ratio of a word is determined, by calculating the probability of a word in the target language and the probability of that word in a non-target language and applying Equation 1.

$$\log_2 \frac{P(w|rightlang) * (1 - P(w|wronglang))}{(1 - P(w|rightlang)) * P(w|wronglang)} \quad (1)$$

But while it is fairly easy to calculate $P(w|rightlang)$ for a language, given a sizable amount of data, $P(w|wronglang)$ is harder to calculate, unless we supply CORPUSCOLLIE with a prohibitively large lexicon of the world’s languages. This is why in the first loop, we restrict CORPUSCOLLIE to selecting seed words on the basis of $P(w|rightlang)$ and the *safetynet* only.

In subsequent loops however, we can use the output of the language identification module, to calculate $P(w|wronglang)$, i.e. the probability that a word occurs in a document that was classified as a non-target language document. Alternatively, if a user knows that his target language **A** has a lot of lexical overlap with another language **B**, (s)he can add documents in language **B** as *wronglang* data to the system, so that appropriate odds ratios can be calculated before the first loop as well.

3 Language Identification

One of the core modules in CORPUSCOLLIE performs language identification. Language identification is a well-studied problem and most techniques use some sort of variant of the TextCat approach⁸, which is based on the text categorization algorithm coined in Cavnar and Trenkle (1994).

⁸<http://odur.let.rug.nl/~vannoord/TextCat>

We are very much obliged to Kevin Scannell, who supplied us with a trigram frequency list for 450 languages. We also used the aforementioned Wikipedia data to construct trigram lists for languages not yet covered by Scannell’s data, resulting in language models for over 500 languages.

Unfortunately, the TextCat approach becomes very slow when so many languages are to be considered. We therefore developed a different, faster classification technique, based on maximum-entropy learning. We discard all of the frequency information and construct a training set where each line lists the class (i.e. the ISO 639-3 code), followed by the top-400 trigrams in that language. This is exemplified for Swahili and Zulu in Figure 2.

To evaluate the language identification module, we constructed a test set of (up to) twenty documents for each of the 242 languages for which a sizable Wikipedia data set exists⁹. Table 1 compares our results to that of the TextCat approach, using the trigram frequency lists, and Google’s language identification module.

Algorithm	Accuracy	CPU time
Maximum Entropy	89%	0.3ms
TextCat	54%	23s
Google	39%	1s

Table 1: Experimental Results for language identification task.

It is clear that the maximum-entropy approach outperforms the alternatives, not only in terms of accuracy, but also in terms of execution time. The low accuracy score for Google is surprising, but it is important to point out that Google only supports 104 languages out of the 240 in the test set. When we compare the maximum-entropy approach to Google’s on languages only supported by the latter, Google scores marginally better (92.8% accuracy vs 92.6%). Google’s approach has the added disadvantage that it uses bandwidth for classification, while the maximum entropy approach performs its classification offline.

TextCat’s low accuracy score is rather surprising,

⁹In the interest of methodologically sound held-out validation, the training data for the classifier was constructed without using documents from the test set.

swa wa_ a_k _wa ya_ _ya na_ a_m _na ka_ _ku ni_ _ka a_n ika ali ati a_w ili kat _ma a_y i_y aka ia_ _kw ana kwa za_ tik la_ i_w i_k ani _ki ish wak ina ha_ a_u li_ _la mba uwa ma_ ini kuw _ni a_h ...
zul aba _ng ulu ing la_ thi hi_ a_n uth wa_ ama nga zin _ba uku ezi ngo izi ban ni_ _ab a_u a_i _uk lo_ a_k ase a_e isi eni we_ oku _iz and esi _ne _ku nge hul ala ant hla na_ khu eth lan imi ela nda ntu olo ...

Figure 2: Training Data for Language Identification. Underscores represent whitespace.

particularly compared to those reported in the literature. Error analysis indicates that TextCat does not seem to scale well to a large array of languages, particularly when many of them are closely related and share distinct orthographic features. The Maximum Entropy approach seems less vulnerable to this effect, as the features seem to function as constraints, rather than as probabilistic cues towards disambiguation.

To the best of our knowledge, CORPUSCOLLIE includes the largest language identification module currently available. Despite the large coverage of the module, it may occur that a user is mining a language that is not yet covered. In such a case CORPUSCOLLIE can construct a new language model on the basis of the initial documents used to bootstrap the web mining process.

4 Case-Study: web mining a Luo corpus

When evaluating web miners, it is inherently impossible to calculate the recall of the system, because there is no way to find out how many target-language documents exist on the world wide web. However, we can evaluate precision, by looking at how many pages in a web mined corpus are written in the target language. In this section we describe a short case study, an empirical, quantitative evaluation of CORPUSCOLLIE as a web corpus mining tool for the resource-scarce language of Luo.

Luo (Dholuo) is a Western Nilotic language spoken by around 5 million people in Kenya, Tanzania and Uganda. It has no official status and can easily be called a resource-scarce language and therefore serves as an ideal case-study to evaluate CORPUSCOLLIE. We decided to work with fairly restrictive user-defined parameters (400 seed words, crawl-depth 5 and 2 loops) to limit the number of documents the Luo native speaker needs to evaluate.

We start off with a collection of Luo documents ($\pm 200,000$ words), once again kindly supplied by

	Total	Mainly Luo	Some Luo	No Luo
Documents	830	292	535	3
%	100	35	64	0.5
Words	410k	212k	197k	46

Table 2: Experimental Results for Luo CORPUSCOLLIE Case-Study.

Kevin Scannell. In the first loop we simply select the most frequent words in this set, not present in the *safetynet*. 400 queries were constructed with these seeds, for which Bing returned 14 Luo pages. Further crawling resulted in another 262 Luo pages. In the second loop (with a new and updated seed list), 77 results were returned by Bing, while 1054 were found through crawling. After near-duplicate detection, 830 documents remained.

We then built a simple evaluation interface that displays each of the 830 documents and offers three evaluation options: (1) pages containing mainly Luo, (2) pages containing mainly another language, but some Luo as well, and (3) pages not containing any Luo.

The results can be found in Table 2. Less than 1% of the documents did not contain any Luo whatsoever. This more than encouraging precision score for CORPUSCOLLIE can be explained by a combination of having a good language identifier, selecting appropriate seed words, crawling and accurate text-processing modules.

5 Conclusion and Future Work

We presented CORPUSCOLLIE, a new web mining tool that incorporates a wide range of text processing and classification modules, in particular a wide-coverage and accurate language identification system. We presented experimental results that underline the capability of CORPUSCOLLIE in web mining a corpus for a resource-scarce language.

We are currently constructing a new evaluation framework, MININET, a closed-world data set of interlinked documents in different languages. This will allow us to evaluate CORPUSCOLLIE and other web corpus miners, not only in terms of precision, but now also in terms of recall, i.e. how many of the documents in the target language that exist in the “closed world” were actually retrieved by the tool.

Furthermore, we will extend our language identification module to not only work on the document level, but also on the sentence level. This will allow us to retrieve target-language data within a multilingual document. This is particularly useful when sourcing data from web forums, where typically a mixture of official and regional languages are used interchangeably.

We aim to publicly release CORPUSCOLLIE and the language identification module through <http://AfLaT.org> as a free and open-source software package in Q2 of 2011. This will hopefully lead to further refinements to the tool through the user community, as well as more extensive experimental results for a larger variety of languages.

Acknowledgments

The second author is funded as a Postdoctoral Fellow of the Research Foundation - Flanders (FWO). The authors wish to thank Naomi Maajabu for her annotation efforts and Kevin Scannell for making his data available to us.

References

- M. Baroni and S. Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. In *Proceedings of LREC-2004*, pages 1313–1316, Lisbon, Portugal.
- M. Baroni and S. Bernardini. 2006. *Wacky! Working papers on the Web as Corpus*. Gedit Edizioni, Bologna, Italy.
- W.B. Cavnar and J.M. Trenkle. 1994. N-gram based text categorization. In *Proceedings of SDAIR-94, the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, USA.
- M. Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual Symposium on Theory of Computing (STOC)*, pages 380–388, Montreal, Canada.
- CorpusCatcher. 2011. *by translate.org.za*. Available from: <http://translate.sourceforge.net> (Accessed: 22 January 2011).
- A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4): Can we beat Google?*, pages 47–54, Marrakech, Morocco.
- A. Finn, N. Kushmerick, and B. Smyth. 2001. Fact or fiction: Content classification for digital libraries. In *Proceedings of the Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*, pages 1–6, Dublin.
- W.H. Fletcher. 2004. Making the web more useful as a source for linguistic corpora. In U. Connor and T. Upton, editors, *Applied Corpus Linguistics: A Multidimensional Perspective*, pages 191–205. Rodopi, Amsterdam.
- R. Ghani and R. Jones. 2000. Automatically building a corpus for a minority language from the web. In *Proceedings of the Student Workshop at the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 29–36, Hong Kong.
- R. Ghani, R. Jones, and D. Mladenic. 2001. Mining the web to create minority language corpora. In *Proceedings of the ACM CIKM International Conference 2001*, pages 279–286, New York, USA.
- N. Ide, R. Reppen, and K. Suderman. 2002. The American national corpus: More than the web can provide. In *Proceedings of the 3rd Language Resources and Evaluation Conference (LREC)*, pages 839–844, Canary Islands.
- A. Kilgarriff, S. Reddy, J. Pomik'alek, and A. PVS. 2010. A corpus factory for many languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC-2011)*, Valletta, Malta.
- J. Prasanna Kumar and P. Govindarajulu. 2009. Duplicate and near duplicate document detection: A review. *European Journal of Scientific Research*, 32(4):514–527.
- L.M. Paul, editor. 2009. *Ethnologue: Languages of the World*. SIL International, Dallas, USA. Online version: <http://www.ethnologue.com>.
- Ph. Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL '99)*, pages 527–534, College Park, USA.
- K.P. Scannell. 2007. The Crúbadán project: Corpus building for under-resourced languages. In *Proceedings of the Third Web as Corpus Workshop: Building and Exploring Web Corpora*, pages 5–15, Louvain-la-Neuve, Belgium.
- S. Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In *Wacky! Working papers on the Web as Corpus*, pages 63–98. GEDIT, Bologna, Italy.

Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages – The Case of Amharic

Martha Yifiru Tachbelie and Solomon Teferra Abate and Laurent Besacier

Laboratoire d’informatique de Grenoble (LIG)

Université Joseph Fourier (UJF)

{martha.tachbelie, solomon.abate, laurent.besacier}@imag.fr

Abstract

This paper presents part-of-speech (POS) tagging experiments conducted to identify the best method for under-resourced and morphologically rich languages. The experiments have been conducted using different tagging strategies and different training data sizes for Amharic. Experiments on word segmentation and tag hypotheses combination have also been conducted to improve tagging accuracy. The results showed that methods like MBT are good for under-resourced languages. Moreover, segmenting words composed of morphemes of different POS tags and tag hypotheses combination are promising directions to improve tagging performance for under-resourced and morphologically rich languages.

1 Introduction

Many languages, specially languages of developing countries, lack sufficient resources and tools required for the implementation of human language technologies. These languages are commonly referred to as under-resourced or pi languages (Besacier et al., 2006). Natural language technologies for these languages are developed using small set of data collected by researchers and, therefore, the performance of such systems are often inferior compared to systems of technologically favored languages. The problem is further aggravated if the language under study is also morphologically rich as the number of out-of-vocabulary (OOV) words is usually big. Therefore, methods that work best with the available resource have to be identified.

In this paper, we present POS tagging experiments conducted to identify methods which result in good performance with small data set available

for under-resourced and morphologically rich languages, taking Amharic as a case. Amharic is one of the under-resourced and morphologically rich languages. It is a major language spoken mainly in Ethiopia and belongs to the Semitic branch of the Afro-Asiatic super family.

The next section presents previous works on Amharic part-of-speech (POS) tagging. In Section 2, we describe the POS tagging methods/software used in our experiments. Section 3 presents the corpus as well as tag-sets used in the experiments and the results of the experiments. Experimental results with segmented data and tag hypotheses combination are given in Sections 4 and 5, respectively. Finally, in Section 6 we render our conclusions and future works.

1.1 Previous Works on Amharic POS tagging

The first attempt in Amharic POS tagging is due to Getachew (2000). He attempted to develop a Hidden Markov Model (HMM) based POS tagger. He extracted a total of 23 POS tags from a page long text (300 words) which is also used for training and testing the POS tagger. The tagger does not have the capability of guessing the POS tag of unknown words.

Adafre (2005) developed a POS tagger using Conditional Random Fields. Instead of using the POS tag-set developed by Getachew (2000), Adafre (2005) developed another abstract tag-set (consisting of 10 tags). He trained the tagger on a manually annotated text corpus of five Amharic news articles (1000 words) and obtained an accuracy of 74%.

Gambäck et al. (2009) compared three tagging strategies – Hidden Markov Models (HMM), Support Vector Machines (SVM) and Maximum Entropy (ME) – using the manually annotated corpus (Demeke and Getachew, 2006) developed at the Ethiopian Language Research Center (ELRC) of Addis Ababa University. Since the corpus

contains a few errors and tagging inconsistencies, they cleaned the corpus. Cleaning includes tagging non-tagged items, correcting some tagging errors and misspellings, merging collocations tagged with a single tag, and tagging punctuations (such as “ and /) consistently. They have used three tag-sets: the one used in Adafre (2005), the original tag-set developed at ELRC that consists of 30 tags and the 11 basic classes of the ELRC tag-set. The average accuracies (after 10-fold cross validation) are 85.56, 88.30, 87.87 for the TnT-, SVM- and ME-based taggers, respectively for the ELRC tag-set.

Tachbelie and Menzel (2009) conducted POS tagging experiments for Amharic in order to use POS information in language modeling. They used the same data used by Gambäck et al. (2009) but without doing any cleaning. TnT- and SVM-based taggers have been developed and compared in terms of performance, tagging speed as well as memory requirement. The results of their experiments show that with respect to accuracy, SVM-based taggers perform better than TnT-based taggers although TnT-based taggers are more efficient with regard to speed and memory requirement. Since their concern was on the accuracy of the taggers, they used SVM-based taggers to tag their text for language modeling experiment.

The present work is different from the above works since its purpose is to identify POS tagging methods that best work for under-resourced and morphologically rich languages. Therefore, different algorithms and different training data sizes have been used to develop POS taggers. Segmentation has also been tried to reduce the effect of morphological feature of the language. Moreover, experiments on tag hypotheses combination have been conducted since it is one way of improving tagging accuracy.

2 The POS Taggers

We have used different tagging strategies in our experiments. This section gives a brief description of the strategies.

Disambig is a module in SRI Language Modeling toolkit (SRILM) (Stolcke, 2010). It translates a stream of tokens from a vocabulary V1 to a corresponding stream of tokens from a vocabulary V2, according to a probabilistic, 1-to-many mapping. Ambiguities in the mapping are resolved by finding the probability $P(V2|V1)$ which is com-

puted as a product of the conditional probabilities $P(V1|V2)$ and a language model for sequences over V2, i.e. $P(V2)$. In our case, V1 consists in word tokens while V2 consists in the corresponding tags. The method has no way to tag unknown words.

Moses is a statistical machine translation (SMT) toolkit that allows to automatically train translation models for any language pair given a parallel corpus (Koehn, 2010). It offers phrase-based and tree-based translation models. In our experiment, the standard phrase-based model has been used and words and POS tags have been considered as a language pair. Similar to disambig, this method does not handle unknown words.

CRF++ is a simple, customizable, and open source toolkit of Conditional Random Fields (CRF) for segmenting/labeling sequential data. CRF++ can be applied to a variety of NLP tasks, such as Named Entity Recognition, Information Extraction, Text Chunking, POS and concept tagging (Lafferty et al., 2001).

SVMTool is a support vector machine based part-of-speech tagger generator (Giménez and Márquez, 2004). As indicated by the developers, it is a simple, flexible, effective and efficient tool.

MBT is a memory-based POS tagger-generator and tagger. The tagger-generator generates a sequence tagger on the basis of a tagged training set and the resulting tagger tags new sequences. Memory-based tagging is based on the idea that words occurring in similar contexts will have the same tag. It is developed using Memory-Based Learning (MBL), a similarity-based supervised learning which is an adaptation and extension of the classical k-Nearest Neighbor (k-NN) (Daelemans et al., 2010).

TnT, Trigram'n'Tags, is a Markov model based, efficient, language independent statistical part of speech tagger (Brants, 2000). It incorporates several methods of smoothing and of handling unknown words. TnT handles unknown words by a suffix trie and successive abstractions while the main smoothing technique used is linear interpolation.

3 Amharic POS Taggers

3.1 The POS tag-set

The POS tag-set developed within “The Annotation of Amharic News Documents” project at the ELRC has been used. The purpose of the

project was to manually tag each Amharic word in its context (Demeke and Getachew, 2006). In this project, a new POS tag-set for Amharic has been derived. The tag-set has 11 basic classes: nouns (N), pronouns (PRON), adjectives (ADJ), adverbs (ADV), verbs (V), prepositions (PREP), conjunction (CONJ), interjection (INT), punctuation (PUNC), numeral (NUM) and UNC which stands for unclassified and is used for words which are difficult to place in any of the classes. Some of these basic classes are further subdivided and a total of 30 POS tags have been identified. Although the tag-set contains a tag for nouns with preposition (NP), with conjunction (NC) and with both preposition and conjunction (NPC), it does not have a separate tag for proper and plural nouns. Therefore, such nouns are assigned the common tag N.

3.2 The corpus

The corpus used to train and test the taggers is also the one developed in the above mentioned project (Demeke and Getachew, 2006). It consists of 210,000 manually annotated tokens of Amharic news documents.

In this corpus, collocations have been annotated inconsistently. Sometimes a collocation is assigned a single POS tag and sometimes each token in a collocation got a separate POS tag. For example, 'tmhrt bEt', which means *school*, has got a single POS tag, N, in some places and a separate POS tags for each of the tokens in some other places. Therefore, unlike Gambäck et al. (2009) who merged a collocation with a single tag, effort has been exerted to annotate collocations consistently by assigning separate POS tags for the individual words in a collocation.

As the tools used for training the taggers require a corpus that lists a word and its tag (separated by white space) per line, we had to process the corpus accordingly. Moreover, the place and date of publication of the news items have been deleted from the corpus as they were not tagged. After doing the pre-processing tasks, we ended up with a corpus that consists in 8,075 tagged sentences or 205,354 tagged tokens.

3.3 Performance of the taggers

The corpus (described in Section 3.2) has been divided into training, development test and evaluation test sets in the proportion of 90:5:5. The development test set has been used for param-

eter tuning and the taggers are finally evaluated on the evaluation test set. We have first trained two taggers using disambig and SMT (moses) on the whole training data. These two taggers do not deal with unknown words. This leads to poor performance (75.1% and 74.4% of accuracy on evaluation test set, for SMT and disambig, respectively) and makes them unpractical for under-resourced and morphologically rich languages. Thus, we decided to experiment on other tagging strategies that have ability of tagging unknown words, namely CRF, SVM, MBT and TnT.

As our aim is to identify methods that best work with small data set and high number of OOV words, we developed several taggers using 25%, 50%, 75% and 100% of the training set. Table 1 shows the performance (overall accuracy as well as accuracy for known and unknown words) of the taggers. We calculated the accuracy gain obtained as a result of using relatively large data by subtracting the accuracy obtained using 25% of the training data from the accuracy we have got using 100% of the training data. Our assumption is that the method with high gain value is dependent on training data size and may not be the best for under-resourced languages.

As it can be seen from Table 1, in most of the cases, increasing the amount of training data resulted in performance improvement. However, the higher increase (gain) has been observed in TnT, which indicates that the performance of this system is more dependent on the size of the training data than the others. This finding is in line with what the TnT developers have said "... *the larger the corpus and the higher the accuracy of the training corpus, the better the performance of the tagger*"(Brants, 2000). Next to TnT, SVM is the second affected (by the amount of data used in training) strategy. On the other hand, MBT has the lowest gain (1.89%), which shows that the performance of MBT is less affected by the amount of data used in training. Daelemans and Zavrel (2010) indicated that one of the advantage of MBT is that relatively small tagged corpus is sufficient for training. The second less affected taggers are CRF-based ones with a gain of 2.37%.

4 Word Segmentation

One of the problems in developing natural language technologies (NLTs) for morphologically rich languages is a high number of OOV words

	Accuracy in %				
	25%	50%	75%	100%	Gain
CRF	83.40	85.01	85.56	85.77	2.37
Kn.	87.16	87.90	88.00	87.92	0.76
Unk.	69.97	70.05	70.07	70.24	0.27
SVM	82.27	83.50	86.16	86.30	4.03
Kn.	85.20	85.67	87.84	87.85	2.65
Unk.	71.80	72.28	75.51	75.10	3.30
MBT	83.54	85.00	85.33	85.43	1.89
Kn.	86.21	87.13	87.12	86.99	0.78
Unk.	74.00	73.95	73.97	74.11	0.11
TnT	79.07	81.77	82.96	83.49	4.42
Kn.	86.44	87.38	87.42	87.60	1.16
Unk.	52.73	52.72	54.71	53.83	1.10

Table 1: Accuracy of taggers on different amount of unsegmented training data.

which leads to poor performance. This problem is more serious for under-resourced languages as the amount of data available for training NLTs is usually limited. A promising direction is to abandon the word as a lexical unit and split words into smaller word fragments or morphemes. This approach is now in use in many NLTs including speech recognition. We have applied such an approach in POS tagging by segmenting words which are assigned compound tags so that the resulting taggers can be applied in sub-word based NLTs.

Since prepositions and conjunctions are attached to nouns, verbs, adjectives, pronouns and even to numbers, compound tags (such as NP, NC, NPC) have been used in the original ELRC tag-set. We segmented prepositions and conjunctions from words and assigned the corresponding tag for each segment. For instance, the word “*läityop’ya*” ‘for Ethiopia’ which was originally assigned the tag NP is segmented into “*lä*” ‘for’ and “*ityop’ya*” ‘Ethiopia’ which are tagged with PREP and N, respectively. Figure 1 shows the rate of OOV words (in the evaluation test set) before and after segmentation for different data sizes. As it can be seen from the figure, the rate of OOV words reduced highly as a result of segmentation. Such an approach also reduced the tag sets from 30 to 16 as it avoids all compound tags which were found in the original ELRC tag-set.

Similar to our experiment described in 3.3, we have developed taggers using different size of the segmented training data. Generally, the taggers developed on segmented data have better accuracy

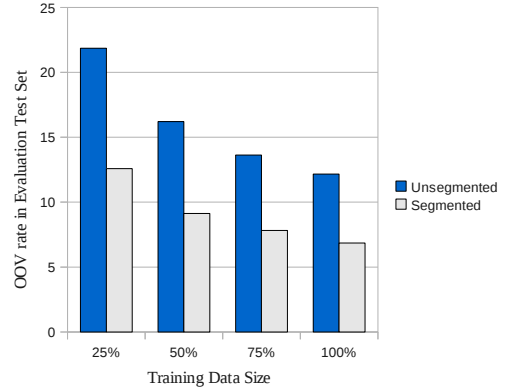


Figure 1: OOV rate before and after segmentation

than the taggers developed using unsegmented data (see Table 2). However, direct comparison of the results is not fair as the tag-set used are different. Although the overall accuracy and the accuracy for known words increased with the training data size, the gain is not as big as it is for the taggers developed on unsegmented data. For all taggers, the accuracy of unknown words decreased as the training data size increases. This is a surprising result which requires further investigation. The result of the experiment further justifies that TnT works better with large training data size (having higher gain, 1.69, compared to the other systems) and MBT is less affected with the amount of training data. The results also enable us to conclude that segmenting words which are composed of morphemes of different POS and which are assigned compound tags is a means of improving tagging accuracy for under-resourced and morphologically rich languages.

	Accuracy in %				
	25%	50%	75%	100%	Gain
CRF	92.39	92.88	93.23	93.42	1.03
Kn.	93.82	93.92	94.19	94.37	0.55
Unk.	82.44	82.47	81.90	80.63	-1.81
SVM	92.64	93.30	93.34	93.50	0.86
Kn.	93.99	94.18	94.21	94.33	0.34
Unk.	83.27	84.57	83.13	82.26	-1.01
MBT	91.45	91.89	91.87	92.13	0.68
Kn.	92.98	92.89	92.79	92.98	0.00
Unk.	80.79	81.95	81.08	80.51	-0.28
TnT	89.98	90.97	91.40	91.67	1.69
Kn.	92.69	93.04	93.24	93.34	0.65
Unk.	71.18	70.38	69.73	68.96	-2.22

Table 2: Accuracy of taggers on different amount of segmented training data.

5 Taggers Combination

Since a possible way of improving POS tagging performance is to combine the output of several taggers, we also experimented on combining the hypotheses of different taggers using four combination methods. As previous experiments [(De Pauw et al., 2006) and (Shacham and Winter, 2007)] on hypotheses combination show that naive approaches outperform the more elaborated methods, three of the combination methods used in our experiments are naive ones. These are majority voting, taking the correct tag from the hypotheses (called as oracle in De Pauw et al. (2006)) and combination of tags proposed for known and unknown words.

In majority voting, as the name implies, a tag that is proposed by most of the taggers is considered as a hypothesis tag for a given word. In case of ties, the tag proposed by the best performing individual tagger is considered. In the oracle combination, among the tags proposed by individual taggers, the one that matches with the gold standard is considered. When no hypothesis matches, the one proposed by the best performing tagger is taken. The third type of combination (called afterword hybrid) is based on the performance of individual taggers for known and unknown words. Our experiment on unsegmented data shows that CRF-based taggers performed best for known words regardless of the size of the training data. On the other hand, MBT- and SVM-based taggers have high performance for unknown words depending on the amount of data used in training (see Table 1). This inspired us to combine the hypotheses of these taggers by taking the tag proposed by CRF-based tagger if the word is known and the tag proposed by MBT-based (for 25% and 50% training set) or SVM-based (for 75% and 100% training set) taggers otherwise.

The fourth combination method is the one applied in speech recognition. This approach (called HPR¹ hereafter) considers the tags proposed by the individual taggers for one word as n-best tags. It, then, generates a confusion network from the n-best and the tag with the highest posterior probability will be selected. Table 3 shows the result of the combined taggers.

As it can be seen from Table 3, on the unsegmented data, majority voting method did not bring improvement over best performing taggers

¹Stands for Highest Posterior Probability.

	Accuracy in %			
	25%	50%	75%	100%
BestTag. ^a	83.54	85.01	86.16	86.30
Majority	83.40	85.00	86.07	86.28
Known	87.16	87.85	87.83	87.84
Unknown	69.97	70.24	74.93	75.02
HPR	85.04	85.82	86.33	86.49
Known	87.44	87.96	87.91	87.92
Unknown	76.43	74.75	76.32	76.17
Hybrid	84.29	85.64	86.30	86.36
Known	87.16	87.90	88.00	87.92
Unknown	74.00	73.95	75.51	75.10
Oracle	88.86	89.35	89.21	89.23
Known	89.96	90.29	89.96	89.83
Unknown	84.92	84.47	84.41	84.83

^aBestTag in Tables 3 and 4 indicates the overall accuracy of best individual taggers.

Table 3: Accuracy of combined taggers on unsegmented data.

for all training data sizes. Moreover, the combined hypotheses of taggers trained on 25% of the data matches with the hypotheses of CRF-based tagger trained on the same data set. This indicates that most of the taggers agree with CRF-based tagger. On the other hand, HPR and hybrid combination methods brought overall performance improvement over the best individual taggers in all the cases. HPR method also consistently improved the accuracy for unknown words. As expected, the oracle approach is the best of all combination method. However, this method is useful only to show the highest attainable performance. Moreover, if the taggers are going to be applied to tag large text (required for language modeling, for instance), the oracle combination method becomes unpractical. Therefore, we can conclude that, the HPR and hybrid combination methods are promising to improve POS tagging performance for under-resourced languages.

For the segmented data, the hybrid method becomes unpractical since SVM-based taggers outperformed all the other taggers in the accuracy of known (except CRF tagger trained on 100% training data) and unknown words regardless of the size of the training data. Therefore, on this data set, only the other three combination methods have been used. As Table 4 shows, the oracle combination shows the best possible performance. The HPR combination outperformed all individual taggers. Like HPR, majority voting resulted in better

performance than all individual taggers but SVM-based taggers trained on 25% of the training data with which it brought the same result.

	Accuracy in %			
	25%	50%	75%	100%
BestTag.	92.64	93.30	93.34	93.50
Majority	92.64	93.31	93.38	93.51
Known	93.99	94.19	94.25	94.34
Unknown	83.27	84.57	83.13	82.26
HPR	92.83	93.36	93.43	93.70
Known	94.05	94.20	94.23	94.44
Unknown	84.35	85.01	84.05	83.55
Oracle	95.06	95.29	95.30	95.40
Known	95.70	95.69	95.69	95.75
Unknown	90.59	91.32	90.70	90.67

Table 4: Accuracy of combined taggers on segmented data.

6 Conclusion

This paper presents POS tagging experiments conducted with the aim of identifying the best method for under-resourced and morphologically rich languages. The result of our POS tagging experiment for Amharic showed that MBT is a good tagging strategy for under-resourced languages as the accuracy of the tagger is less affected as the amount of training data increases compared with other methods, particularly TnT.

We are also able to show that segmenting words composed of morphemes that have different POS tags is a promising direction to get better tagging accuracy for morphologically rich languages. Our experiment on hypothesis combination showed that HPR and hybrid combination methods are practical to bring improvement in tagging under-resourced languages.

In the future, we will apply the taggers in automatic speech recognition as well as statistical machine translation tasks for under-resourced and morphologically-rich languages.

7 Acknowledgment

We would like to thank Bassam Jabaian for his technical help on this work.

References

L. Besacier, V.-B. Le, C. Boitet, V. Berment. 2006. ASR and Translation for Under-Resourced Languages. *Proceedings of IEEE International Confer-*

ence on Acoustics, Speech and Signal Processing, ICASSP 2006, 5:1221–1224.

Mesfin Getachew. 2000. *Automatic Part of Speech Tagging for Amharic Language: An experiment Using Stochastic HMM*. Addis Ababa University, Addis Ababa, Ethiopia.

Sisay Fissaha Adafre. 2005. Part of Speech Tagging for Amharic using Conditional Random Fields. *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 47–54.

Björn Gambäck, Fredrik Olsson, Atelach Alemu Argaw, Lars Asker. 2009. Methods for Amharic Part-of-Speech Tagging. *Proceedings of the EACL Workshop on Language Technologies for African Languages - AfLaT 2009*, 104–111.

Girma Awgichew Demeke and Mesfin Getachew. 2006. Manual Annotation of Amharic News Items with Part-of-Speech Tags and its Challenges. *ELRC Working Papers*, 2(1):1–17.

Martha Yifiru Tachbelie and Wolfgang Menzel. 2009. Amharic Part-of-Speech Tagger for Factored Language Modeling. *Proceedings of the International Conference RANLP-2009*, 428–433.

Andreas Stolcke. 2002. SRILM — An Extensible Language Modeling Toolkit. *Proceedings of International Conference on Spoken Language Processing*, 2:901–904

Philipp Koehn. 2010. Moses - Statistical Machine Translation System: User Manual and Code Guide. Available from: <http://www.statmt.org/moses/manual/manual.pdf>.

John Lafferty, Andrew McCallum, Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*, 282–289.

Thorsten Brants. 2000. TnT — A statistical Part-of-Speech Tagger. *Proceedings of the 6th ANLP*.

Jesús Giménez and Lluís Màrquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Walter Daelemans, Jakub Zavrel, Antal van den Bosch, Ko van der Sloot. 2010. MBT: Memory-Based Tagger, Version 3.2, Reference Guide. ILK Technical Report ILK 10-04 Available from: <http://ilk.uvt.nl/downloads/pub/papers/ilk.1004.pdf>

Walter Daelemans and Jakub Zavrel. 1996. MBT: A Memory-Based Part of Speech Tagger-Generator. In *Proceedings WVLC*

Guy De Pauw, Gilles-Maurice de Schryver, Peter Wagacha. 2006. Data-Driven Part-of-Speech Tagging of Kiswahili. *Proceedings of the 9th International Conference on Text, Speech and Dialogue*, 197–204.

Danny Shacham and Shuly Winter. 2007. Morphological disambiguation of hebrew: A case study in classifier combination. In *Proceedings of Empirical Methods in Natural Language Processing*, 439–447.

Language Resources for Mongolian

Jaimai Purev

Center for Research on Language Processing,
National University of Mongolia
Ulaanbaatar-210646, Mongolia
purev@num.edu.mn

Chagnaa Altangerel

Center for Research on Language Processing,
National University of Mongolia
Ulaanbaatar-210646, Mongolia
altangerel@num.edu.mn

Abstract

Mongolian language is spoken by about 8 million speakers. This paper summarizes the current status of its resources and tools used for Mongolian language processing.

1 Introduction

As to its origins, the Mongolian language belongs to the Altaic language family, and typologically, it is an agglutinative language. Mongolian is the national language of Mongolia and today it has about 8 million speakers around the world including Mongolia (2.7 mln), Inner Mongolia in China (3.38 mln), Afghanistan (?) and Russia (0.5 mln) (Gordon 2005, Purev 2007). Nowadays, Mongolian is written in two official scripts: the Cyrillic Mongolian Script and the (old, uigur) Mongolian Script. The old Mongolian script was used before introduction of the Cyrillic script in 1940s. But, the Cyrillic Mongolian is predominately used in everyday life and also on the Internet. Thus, it is used also for most of research on local language processing and resource development.

Today, manually produced resources such as dictionary are available in Mongolian, and their use is closed to research purpose. The resource such as text and speech corpora, tagged lexicon and a large amount of dictionary in digital content are needed for further Mongolian language processing.

The paper focuses on the Mongolian language resources and tools developed, which can be used for research in computational linguistics and local language processing.

2 Mongolian Language Resources

Mongolian is a less developed language in the computer environment. There have been very few digital resources and research work for it. Recently, several research works aiming to develop local language resources, which are a 5 million word text corpus, a 3 thousand word speech corpus for Mongolian speech recognition, and a 1,500 sentence speech corpus for text to speech training, have begun, respectively. In last few years, some research projects such as Mongolian Text To Speech (TTS) (InfoCon, 2006) and Mongolian script converter (Choimaa and Lodoisamba, 2005) have implemented. Currently, these projects are inactively used in research and public usage.

In the following chapters we will introduce Mongolian language resources and various tools developed till now.

2.1 Text Corpus for Mongolian

Center for Research on Language Processing (CRLP) at the National University of Mongolia (NUM) in Mongolia has been developing corpus and associated tools for Mongolian. A recent project collected a raw corpus of 5 million words of Mongolian text mostly from domains like daily online or printed newspapers, literature, and laws. This corpus was reduced to 4.8 million words after cleaning and correcting some errors in it. The cleaned corpus comprises 144 texts from laws; 278 stories, 8 novelettes, and 4 novels from literature; 597 news, 505 interviews, 302 reports, 578 essays, 469 stories, and 1,258 editorials from newspaper, respectively. The domain-wise figures are given in Table 1.

Domains	Cleaned Corpus	
	Total Words	Distinct Words
Literature	1,012,779	78,972
Law	577,708	15,235
publish	2,460,225	118,601
Newspaper “Unen Sonin”	949,558	61,125
Total	5,000,270	192,061

Table 1. Distribution of Mongolian Corpus

In this corpus 100 thousand words have been manually tagged. As building the corpus for Mongolian, we have developed other resources such as part of speech (POS) tagset, dictionary, lexicon, etc based on the corpus.

2.2 Speech Corpus

Based on the previous 5 million word corpus 1500 phoneme balanced sentences are selected and built speech corpus. This corpus is labeled and used for training Mongolian TTS engine. Beside the corpus, there are also 20 vowel phonemes and 34 consonant phonemes (which make total 54 phonemes) identified.

In mission to develop the Mongolian speech recognition system a general isolated-word recognizer for Mongolian language (native 5 male and 5 female speakers, 2500 isolated words dictionary) has been constructed with using the HTK toolkits (Altangerel and Damdinsuren 2008). Its recognition accuracy is about 95% on isolated word basis.

2.3 Machine readable dictionaries and the-saurus

An English-Mongolian bilingual dictionary which is based on Oxford-Monsudar printed dictionary has about 43K headwords which have about 80K senses. Each head word has its POS tag and each sense has its appropriate key words for collocation and/or senses. Those keywords are used for disambiguating word senses in English-Mongolian machine translation system.

HeadID	SenseN	Sense	SenseTra	SenseColl
101732	10173201	enquire as to	acyyx	name, reason
101732	10173202	request	хыгсах	permission, toler.
101732	10173203	invite	уух	person
101732	10111801			

Figure 1. An entry in En-Mon dictionary

Secondly, there is a corpus of digitized Mongolian monolingual dictionary of 35K entries,

with their corresponding examples. From these entries about 10K nouns were selected and created a hierarchy of hyponymy via manual and semiautomatic method (the tool is introduced in the following subsection).

ID	word	sense1	sense2	explain
47369	1140 ХАРУУЛ	I	1	манав, манаж харгалзах этгээд;
47370	1140 ХАРУУЛ	I	2	холын барааг харахаар байгуулсан өндөрлөг тагт.
47371	1141 ХАРУУЛ	II		хил хязгаар, боомт газрыг сэргийлж хамгаалах алба;
47372	1142 ХАРУУЛ	III		моддыг харуудан засах багаж;

Figure 2. Entries in the monolingual dictionary

Additionally, there is an English-Mongolian idiom dictionary created from frequently used idioms with about 2K entries. In addition to English idiom and corresponding Mongolian translation, each entry has a regular expression to identify it in a sentence.

to toe the party line	<code>\b(toe toes toeing toed)\b the party line</code>	навын дэг жэгийг ягтал баримтлах, намын ёс журмыг ягтал баримтлах
to touch a raw nerve	<code>\b(touch touches touching touched)\b a raw nerve</code>	эмзэг газрыг нь олж хатгах
to touch nerve	<code>\b(touch touches touching touched)\b nerve</code>	эмзэг газрыг нь хөндөх
to treat sb with kid gloves	<code>\b(treat treats treating treated)\b (.*) with kid gloves</code>	хүний эвийг олох
to turn in one's grave	<code>\b(turn turns turning turned)\b in (.*)'s his my her our their) grave</code>	яс нь өндөлзөх

Figure 3. Some entries in an idiom dictionary

For example, to recognize the all possible forms of the idiom *to toe the party line* we have set the regular expression as `\b(toe|toes|toeing|toed)\b the party line`, and since it is a verbal phrase, we have selected the verb *toe* as a main constituent. In further applications such as machine translation, this idiom is seen as a verb and its tense is reflected in its main constituent *toe* as *toes*, *toed*, *toeing* etc. In more detail, the sentence *He toed the party line after he joined the club* is simplified to *He toed* after he joined the club*. Analyses such as POS tagging and/or syntactic parsing will be simpler afterward, since the whole idiom is replaced with only one constituent (a verb in this example) without any change of the sentence constituents. After analyzing the simplified sentence the translation of the idiom is put into the verb *toed**, which is in the past tense.

Beside those dictionaries a smaller size dictionary of proper noun and abbreviation is also compiled and is being enriched.

2.4 Tools

There are also additional tools available through CRLP for text normalization, dictionary based spell checking, POS tagging and word, sentence segmentation, word syllabication etc. These will be introduced in the following parts.

Spell Checker

We have developed a dictionary-based spell-checker for cleaning a Mongolian corpus. The overall system is comprised of a user GUI, a word speller, a word corrector, a lexicon and corpora as shown in the following figure.

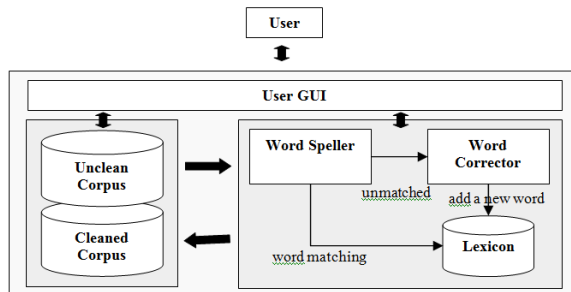


Figure 4. Overall architecture of Mongolian Spell-checker

Word speller is an important part of the system. It will perform an operation of checking an input word that is given by a user. For doing that, the input word is compared to the words in a dictionary the speller contains. The dictionary is loaded into memory from a lexicon, currently containing around 100 thousand words. Its data structure is a tree shown in the following figure.

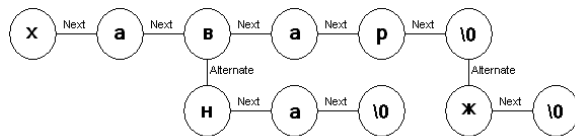


Figure 5. Tree Structure of Dictionary

Here is an experiment that shows the effective memory usage and data structure. For loading 24,385 headwords whose characters are 184,777 in total into the memory, the number of tree nodes allocated into memory is 89,250, or two times less than that of the actual words characters. Thus, this kind of tree structure is suitable to storing a large amount of words and retrieving a word from them in a short time.

Word corrector is a part to correct an input word if it is not found in the dictionary, or could be misspelled. This module of the system finds possible correct forms of the input word by matching it with the dictionary used in the word speller. A correcting algorithm considers four kinds of misspelling errors. First, one of the letters in a word is mistyped, for example, a word *father* can be mistyped as *tather*. Second, a letter is mistakenly inserted into a word, for example, *father* can be mistyped as *fatherr*. The third error

type is that one letter of a word is missed, for example, *father* can be mistyped as *fathr*. Last, two letters of a word can be exchanged, for example, *father* can be mistyped as *fahter*. The algorithm is potential to correct a wrong word whose up to three letters are misspelled.

The spell checker needs more development in the future. Since it uses only a dictionary, that is a simple list of words, it automatically corrects words without accounting about the context in which the word occurs. Sometimes it leads to a problem when working on a optically character recognized (OCR'd) texts. OCR'd texts are not mistyped words but they are misrecognized words and in this case spell checker needs to be more intelligent.

Unicoded PC-KIMMO and two level rules

History of using a finite state tool (FST) in Mongolian morphology begins earlier [Purev et al, 2007]. Currently, for processing Mongolian words, generator and recognizer part of famous morphological parser PC-KIMMO was updated to support Unicode. Besides this, Mongolian morphological rules were compiled in 84 two level rules. These rules account the vowel harmony of the Mongolian language, which was not done in previous attempts. The system generates and recognizes with high level of accuracy. It achieves about 98 percent on 1,000 words covering all the types (29 classes for nouns, 18 classes for verbs) in Mongolian morphological dictionary. In the future system it needs to include word grammar files.

Text normalization tool

Since the collected texts are stored in digital form, they have two problems which are mixed character encoding and miss-file encoding. Cyrillic characters are encoded in either American Standard Code for Information Interchange (ASCII, Windows 1251 encoding) or Unicode. Thus some files contain texts written in both encodings. Also some UTF files are saved in normal .txt files even though they contain Unicode texts. That is why we developed a tool, named "Text file Converter", for changing the files encoded with Unicode to UTF8, and for changing ASCII characters into Unicode ones. This tool first checks file's encoding and converts to UTF. Afterwards it checks mixed character encodings and fixes them. We have developed these tools from scratch instead of using existing converter, because we needed to do fixing of mixed encoding within a file, but existing converter tools mainly converts between homogenously encoded files.

Mongolian raw corpus has some problems needed to be cleaned, such as: ASCII and Unicode mixture, Interchanged characters: Characters with similar shapes are used interchangeably, Latin and Cyrillic characters used in a same word, Letters in number, such as ‘o’ is used as ‘0’ in numbers: “35oo”, Characters located closely on the keyboards are mistyped, longer words separated by hyphen(s), Quotations used asymmetrically, Character order is changed, Words combined without delimiters, Misspelled words etc.

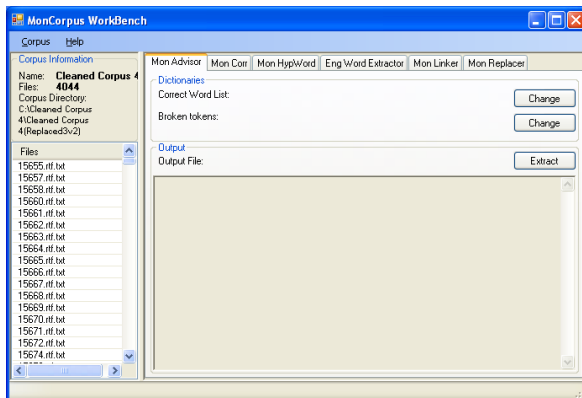


Figure 6. A Mongolian corpus normalizing tool, “MonCorpus Workbench”

To remove those we have developed Mongolian Corpus Workbench, a set of semi-automatic cleaning and normalizing tools.

POS Tagger

Incorporated knowledge of the language in corpus is used as the main data for the development and application of the corpus-based systems. Specially, part of speech tag is a key to the language processing.

Initially, manual POS tagger, followed by unigram, bigram and trigram taggers are developed.

The manual tagger, named MTagger, is used for tagging the text in the XML format. In a corpus building, there is a need to analyze raw text and tagged text except tagging, and we also needed the tool for such purpose. Therefore, we developed some text analyzing tools such as searching, filtering and computing statistical information, and plugged in MTagger.

POS Tagset editing: The user can edit the POS tagset freely during the tagging and can create own POS tagset. The tagset is designed to be in a common text file. The file format of POS tagset is very easy to understand and each line of a tagset file contains individual tag information that consists of tag and its description. Also some

tags are grouped into one group as a ">,separator" line that must be in below position of the last tag of the group tags.

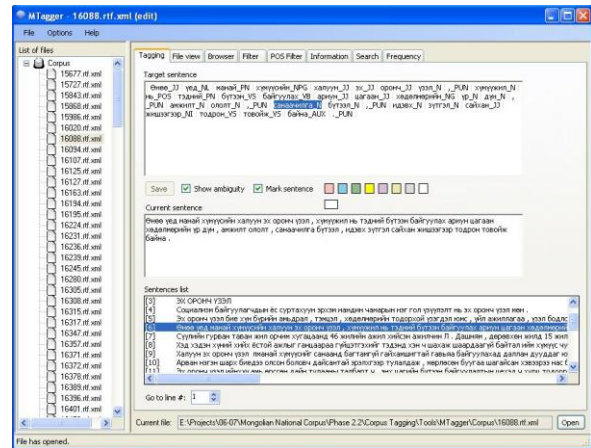


Figure 7. Manual tagger Mtagger.

POS tag suggestion: POS tag suggestion helps users to assign a corresponding POS tag to a word and lists appropriate POS tags based on corpus words and tags frequency. If you have created list of word frequency from the corpus, MTagger will suggest appropriate POS tags when you are assigning a tag.

Ambiguity tag bookmarks: Users may encounter ambiguity case because of one or more tags may seem to be assigned a given word. If users are not sure to assign POS tag for selected word or they cannot choose one of believable POS tags, they can use bookmarks for that. In fact, users should use ambiguity POS tag "?" in such situation, then attaches ambiguity POS tags for selected word. After that, users can search bookmarked words.

Auto-completion tool: The auto-completion tool lets you choose POS tags to auto-complete names for you while documents are being tagged. This functionality requires accessibility to each of the POS tags you want to use. To enable auto-completion, select each of a word that you want to use auto completion in the auto-completion control. This ability allows users to tag documents without mouse.

Searching word in documents: Users can search word only from a document that is being tagged. Mostly, the searching function is needed when user is editing a previously tagged file and the searching function highlights the words to be searched in the file. Then, user can select a sentence with searching word and view on Tagging field. There are two searching options; match case and match whole word.

Word comparison by POS tag: MTagger compares words by POS tags from a working file or whole corpus. The limitation of MTagger is users can select at most 5 POS tags for comparing words. Shown in a table, result of comparing can be printed. The result of comparing shows the words and their frequencies. The first column of the result is words, second is their frequencies. A total number of words and a total number of tags are shown in the bottom of each result.

Searching word using XPath: It is possible to search using XPath expression only from a selected file. By entering a POS tag in POS field and a suffix tag in Suffix field, MTagger creates automatically XPath expression. Or, user who has enough knowledge of XPath expression can write easily in XPath field. Selecting words with options locate in file view or tagged view from the result, a selected word is located either editing file field or tagging field.

Word frequency: As counting word frequency, it creates corpus statistic information and list of word frequency file which is created from all of corpus files, with stat extension. The name of stat file is same as corpus name that it contains information about how many parts-of-speech tags and how often it is tagged for each word. After counting word frequency, POS suggestion function of MTagger will be activated.

Statistical information: MTagger shows neither selected file nor Corpus statistic information. For file information, paragraphs, sentences and tagged words are included and for corpus information, number of files, number of tagged files, tagged words (tokens) and distinct tagged words (word types) are included.

Because of the statistical issue occurring in insufficient training data, in a trigram model we have also taken unigram and bigrams into account. This method was first used in TnT tagger [Brants, 2000] and known as one of the widely used methods in the field. The trigram tagger was trained on 450 sentences which includes 60K words and tested on the 10K word texts of various contents. The following table shows the accuracy of the trigram tagger.

Text	Text 1	Text 2	Text 3	Average
#Words	10,390	11,858	11,000	11,083
OOV, %	3.2	14.6	19.6	12.5
GuessEnd, %	40.3	45.6	26.4	37.4
TotalAcc, %	95.8	90.3	83.3	89.8

Table 2. Accuracy of trigram tagger

Out of vocabulary (OOV, not trained) words in the test set was about 13 percent (OOV col-

umn) and the tagger guessed tags (GuessEnd column) based on the word endings. It is shown that as the number of OOV word increases the accuracy of the tagger decreases. It mainly depends on the length of the endings used in the guessing algorithm.

Manual and Semiautomatic WordNet Creating Tool for Mongolian

Manual editor for Mongolian lexical semantic network (LSN) is developed in VS #C [Altangerel, 2010]. User interface shows Network in a tree structure, Vocabulary or word sense repository, Detail fields, editing section. Also it has functions of filtering entries, Adding new nodes by drag and drop from entry list to hierarchy/tree via mouse, Editing LSN /by right mouse click: Adding, Removing, Editing a node.

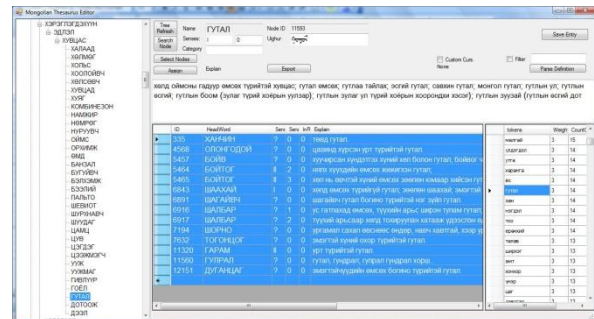


Figure 8. Semi-automatic tool for Mongolian lexical semantic network

A semiautomatic tool has additional modules for clustering, definition parsing etc. It uses some language specific heuristics to select the feature of the word from its definition in monolingual dictionary [Altangerel and Cheol-Young, 2010]. User can select a particular cluster and remove some entries from it and assign it to the network.

Syllabication and phoneme tools

In the framework of Mongolian TTS development, syllabication tool and phoneme frequency counter are built.

Transfer rule editing tools

For building rule based machine translation system, transfer and generation rule editing tools have been developed in Java.

The generation tool retrieves some patterns from the parsed corpus based on Tregex (tree regex) [Roger and Galen, 2006] patterns and does some operations (remove, insert, move, rename etc) on the tree.

Additional to the pattern search user can also use dependency information in the query. With this tool we have created about 300 generation

rules for English Mongolian machine translation system.

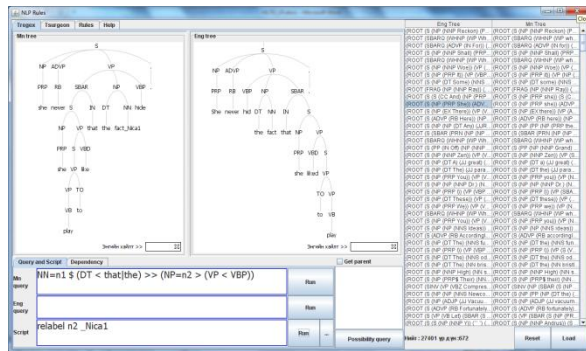


Figure 9. Generation rule editor

Transfer rule editing tool searches a phrase structure grammar from sentences in the Penn Treebank corpus and allows editor to set the constituent order into target language. We have set the transfer order (from English to Mongolian) for the most frequent 5K rules from Penn Treebank.

English-Mongolian Machine Translation Engine

Finally, rule based English-Mongolian machine translation engine should be introduced here. The engine uses most of the resources mentioned above and translates English sentence into Mongolian.

It is mainly developed in Java, based on open source tools and put in public use via web interface. More information about this engine is given in a separate paper.

3 Conclusion

This paper lists some core linguistic resources of Mongolian and related tools, available through CRLP and other sources. For dissemination it needs to be addressed.

The resources and tools in Mongolian language are being used in systems such as Mongolian text to speech engine, English to Mongolian machine translation system etc.

We have created the tools mainly from scratch, instead of using existing tools, because of some peculiarities in Mongolian language, and also of some requirements of the data origin.

As our experience in relating to develop Mongolian language processing for years, we have faced some situations challenging us. They are that how to improve local people interests in Mongolian language processing, lack of researchers and staffs who have experience and knowledge on language processing and government issues to support local language processing.

4 Acknowledgement

Some work described in this paper has been supported by PAN Localization Project (PANL10n), ITU-AMD and ARC.

TTS for Mongolian project is undergoing and it is supported by ITU-AMD with cooperation with NECTEC, Thailand.

Reference

Altangerel, A. and Damdinsuren, B. 2008. *A Large Vocabulary Speech Recognition System for Mongolian Language*. the Proceedings of Oriental CO-COSDA 2008 Workshop, Koyoto, Japan

Altangerel Chagnaa. 2010. *Lexical semantic network for Mongolian*, Khurel Togoot national conference proceeding, pp 207-210, Ulaanbaatar, Mongolia.

Altangerel Chagnaa and Cheol-Young Ock. 2010. *Toward Automatic Construction of Lexical Semantic Networks*, 6th International Conference on Networked Computing (INC), 2010, ISBN: 978-1-4244-6986-4

Brants T. 2000. *TnT – a Statistical Part-of-Speech Tagger*, in Proc. sixth conference on applied natural language processing (ANLP-2000), Seattle, WA, 2000.

Choimaa Sch. and Lodoisamba S. 2005. *Mongol hel-nii tailbar toli* (Descriptive dictionary of Mongolian).

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical NLP*, MIT Press.

Daniel Jurafsky, James H. Martin. 2000. *Speech and Language Processing*, Singapore.

Gordon, Raymond G., Jr. (ed.). (2005). *Ethnologue: Languages of the World*, Fifteenth edition. Dallas, Tex.: SIL International.

InfoCon Co., Ltd, TTS for Mongolian: <http://www.infocon.mn/tts/>

Purev Jaimai, Tsolmon Zundui, Altangerel Chagnaa, and Cheol-Young Ock. 2007. *PC-KIMMO-based Description of Mongolian Morphology*. International Journal of Information Processing Systems, Vol. 1 (1), pp. 41-48.

Purev Jaimai. 2007. *Linguistic Issues in Mongolian Language Technology*. In the Proceedings of 1st Korea-Mongolia International Workshop on Electronics and Information. Chungbuk national University, Cheongju, Korea.

Roger Levy and Galen Andrew. 2006. *Tregex and Tsurgeon: tools for querying and manipulating tree data structures*. 5th International Conference on Language Resources and Evaluation (LREC 2006)

Dzongkha Phonetic Set Description and Pronunciation Rules

Dechen Chhoeden, Uden Sherpa, Dawa Pemo, Pema Choejey

Department of Information Technology and Telecom, Bhutan

{dchhoeden, usherpa, dpemo, pchoejey} @dit.gov.bt

Abstract

This paper describes the Dzongkha phonemic inventory that was defined during the Text Analysis process for Dzongkha Text-to-Speech System (TTS) prototype developed in collaboration with NECTEC, Thailand (Sherpa et al., 2008). About 56 unique phonemes were discovered covering four different classifications, viz., initial consonants, vowels, diphthongs and consonant clusters. These phonemes were transcribed using equivalent Latin representations. In the phonemic set, two tones were identified: first is the normal tone and the second one is the modification of the normal tone. The modified tone can be identified by a high or sharp tone. Following the importance of pronunciation of syllables and words, and its associated rules in TTS systems, some basic syllable-level pronunciation rules are also explored.

1. Introduction

Dzongkha is the official and the national language of Bhutan. It belongs to the Sino-Tibetan family of languages. The script used in writing Dzongkha is identical to the Tibetan script and is known as ‘Uchen’ script (van Driem and Tshering, 1998). The spoken form of Dzongkha is not only different from Tibetan but also the vocabulary set used are different from each other leaving very few exceptions. It can be rightly said that the relationship between the spelling of the written language and the actual pronunciation while speaking is not straightforward for both the languages as described by van Driem (1992).

In the phonetic set definition, it was found that spoken Dzongkha could be represented by 30 initial consonants, 5 clusters, 11 vowels and 10 diphthongs. Each of the phonemes are classified and defined along with some examples.

2. Phonemic Inventory Design

Each of the unique phonemes of the language is transcribed using relevant Latin string (Romanization) equivalent, as demonstrated in the following:

a) Initial Consonants: Thirty initial consonants can be accounted for in Dzongkha language. Table 1 depicts the list of thirty consonants in the traditional Dzongkha alphabetical order. Note that in the given example syllables, '0' indicates normal tone & '1' represents the high tone as described in part f) of Section 2.

IPA	Dzongkha Alphabets	Transcribed Phoneme Equivalent	Example Script	Romanized	Transcription based on the pronunciation
K	ཀ	K	དཀར་	d k r	k a r 0
Kh	ཁ	Kh	མཁར་	m kh r	kh a r 0
G	ག	G	དག་	d g hh	g a 0
N	ང	Ng	ངག་ སྒྲ་	ng g s ng n	ng a g 0 ng e n 1
tS	ཅ	C	བཅག་	b c g	c a g 0
tSH	ཆ	ch	ཚོག་	ch o g	ch o g 0
dZ	ཇ	j	མཇལ་	m j l	j e l 0
Ø	ཉ	ny	ཉལ་ སྟླ་	ny l s ny n	ny e l 0 n e n 1
T	ཏ	t	བཏར་	b t ng	t a ng 0
tH	ཐ	th	ཐབ་	th b	th a b 0
D	ད	d	གདོང་	g d o ng	d o ng 0
N	ན	n	ནོར་ སྟུམ་	n o r s n m	n o r 0 n a m 1
P	པ	p	པར་	p r	p a r 0
pH	ཕ	ph	ཕགས་	ph g s	ph a g 0
B	བ	b	བབ་	b b	b a b 0
M	མ	m	མིག་ མེགས་	m i g r m g p	m i g 0 m a p 1
Ts	ཅ	ts	གཅོང་	g ts ng	ts a ng 0
tsH	ཆ	tsh	ཚོལ་	tsh l	tsh e l 0
Dz	ཇ	dz	འཇོམས་	hh dz o m s	dz o m 0
W	མ	w	མ་ མང་	w w ng	w a 0 w a ng 1

Z	ཞ	zh	གཞུང་	g zh u ng	zh u ng 0
z	ཟ	z	ཟམ་	z m	z a m 0
ú	ཨ	hh	ཨོངས་	hh o ng s	hh o ng 0
j	ཡ	y	ཡལ་	y l	y e l 0
			གཡོག་	g y o g	y o l
..	ར	r	རིལ་	r il	r il 0
l	ལ	l	ལེན་	l e n	l e n 0
S	ཤ	sh	ཤིང་	sh i ng	sh i ng 0
s	ས	s	སསངས་	b s ng s	s a ng 0
h	ཧ	h	ཧང་	h ng	h a ng 0
/	ཨ	@	ཨམ་	@ m	@ a m 0

Table 1. Initial consonant Inventory

The above table shows the transcription of the 30 initial consonants. Their corresponding phonetic representation is the equivalent Latin string counterparts. It is to be noted that the consonants contains the inherent vowel 'a'. Therefore, when there is no particular vowel, the vowel 'a' is appended to the root consonant of a syllable or word.

For example, the consonant letter ། is pronounced as 'k a 0' with an inherent vowel 'a', instead of a simple 'k'.

b) Vowels: Spoken Dzongkha has eleven vowels, including five basic vowels. See Table 2 below.

IPA	Vowel	Transcribed Phoneme Equivalent	Example Transcription Script	Romanized
a		a	ཨམ་ @ m	@ a m 0
i	ི	i	ཨིན་ @ i n	@ i n 0
u	ུ	u	གུ g u	g u 0
e	ེ	e	དེ d e	d e 0
o	ོ	o	ཨོམ་ @ o m	@ o m 0
ue		ue	གཡུལ་ g y u l	u e l 0

oe		oe	ཨོད་ y o d	y o e 0
aa		aa	ཨཱ་ a h h	@ a a 0
ii		ii	ཨིི་ n h h i	n i i 0
uu		uu	ཨུུ་ b s d u w	d u u 0
oo		oo	ཨོོ་ b o l	b o o l 0

Table 2. Spoken Dzongkha vowel inventory

The vowels (excluding 'a') are written as diacritics, above or below the root letter or the main consonant letter. While pronouncing the vowels: **a** is pronounced like 'but', **i** is pronounced like 'fit', **e** is pronounced like 'May', **u** is pronounced like 'food' ().

Pronunciation of some vowels are modified when the root letter is combined with certain suffixes. Only a few number of suffixes and prefixes are available. While the prefixes before the root letters are not at all pronounced, certain suffixes are pronounced along with the root letters. And a handful of them modify the vowels associated with the root letter. Detail of vowel modification is given in Table 3 below.

V	Suffix										
	g	ng	n	b	m	r	l	p	d	s	h
o											
w											
e											
l											
a			e n				e l		e	e	
i											
u			ue n				ue l		ue	ue	
e											
o			e n				e l		e	e	

Table 3. Vowel Modification Table

The table depicts eleven suffixes. Suffixes **d**, **s**, & **hh** are not pronounced in a syllable while the rest are all pronounced. Very few vowels can be modified when combined with particular suffixes. Therefore suffixes **n**, **l**, **d** and **s**, with the vowels **a**, **u** and **o**, will modify the vowels in question.

For example, 'p l' is pronounced as 'p e l 0', not 'p a l 0' where by the vowel 'a' is modified to 'e' vowel.

c) **Diphthongs:** A total of ten diphthongs were identified for spoken Dzongkha as shown in Table 4.

IPA	Diphthong	Example Romanized Transcription Script
ai	ai	ཨའི་ @ hh i @ ai 0
au	au	ཨའུ་ @ hh u @ au 0
ae	ae	ཨའེ་ @ hh e @ ae 0
ui	ui	བུའི་ b u hh i b ui 0
oi	oi	མའི་ m o hh i m oi 0
ou	ou	ལྷོ་ l t o w t ou 0
eu	eu	རེུ་ r e hh u r eu 0
ei	ei	ཐེའི་ s d e hh i d ei 0
eo	eo	ཆེ་ ch e w ch eo 0
iu	iu	ཚེུ་ tsh i hh u tsh iu 0

Table 4. Mapping of Dzongkha Diphthongs

Note: It might be possible to find more diphthongs or vowels, for that matter, in the event of 'the growing text corpus' and its subsequent text analysis.

d) **Clusters:** Spoken Dzongkha has the following five consonant clusters. They are written with special letter conjuncts. For example, in cluster འྲ , letter 'r' is placed under the root letter 'g'.

IPA	Consonant Cluster	Example Romanized Transcription Script
d ^ː	dr	དྲ་ d g r dr a 0
t ^ː	tr	ཐྲ་ s k r tr a 0
tH ^ː	thr	ཐྲ་ th r thr a 0
lh	lhh	ལྷ་ l h lhh a 0
hr	hr	མྲ་ h r hr a 0

Table 5. Mapping of Dzongkha Clusters

Note: The letters are stacked over each other, i.e, the first (root) letter is usually stacked over the second letter.

e) **Final Consonants:** In Dzongkha there are only eight final consonants, see Table 6. They are the suffix letters that are actually pronounced along with the root consonant or letter.

IPA	Computer	Example Romanized Transcription Script
g	g	དག་ d g d a g 0
ng	ng	སྟང་ th u ng th u ng 0
n	n	དོན་ d o n d e n 0
b	b	སྟབས་ s t b s t a b 0
m	m	འཚམ་ hh ch m ch a m 0
R	r	འཇུར་ hh j u r j u r 0
l	l	འཇལ་ ng l ng e l 0
p	p	གསཔ་ g s r p s a p 0

Table 6. Mapping of final consonants

f) **Tones:** Two tones were identified for the language, see Table 7 below. The low tone is the normal tone while the high tone (sharp) is the modification of the former tone. They are represented by '0' and '1' respectively. The high tone variation usually depends on the following:

- Combination of certain prefixes with the root letter
- Combination of certain head letter with the root letter (superscribed)
- The subjoined letter stacked under the root letter (subscribed)

But only a few head letters and a few subjoined letters when combined with particular root letters, will modify the tone of a syllable.

For example, the syllable འྲ་ is pronounced as 'n a ng 0' with normal tone. But by adding the head letter 's' to the the letter འྲ , the syllable is modified to 'n a ng 1'. འྲ་ is pronounced with a high tone.

Example	Romanized	Transcription
བང་	b ng	b a ng 0
དབང་	d b ng	w a ng 1
ནང་	n ng	n a ng 0
སྤྲང་	s n a ng	n a ng 1
བམ་	b m	b a m 1
ལྷམ་	l m	l a m 1

Table 7. Example syllables with Tones identified in Dzongkha Language

Note: In general, it can be noted that the combination of certain letters with the root letter affect the tone of the whole syllable or word.

3. Elements of a Dzongkha Syllable

A syllable is the basic unit of meaning or morpheme in Dzongkha. Syllables are normally delimited by a delimiter " | " known as the 'tsheg' bar or delimited by other punctuation. Because of this syllable delimiter there are no inter-word spaces in Dzongkha. One or more syllables form words. Each syllable contains a root letter (ming-zhi) and may additionally have any/or all of the following parts in the given order: prefix; head letter; sub-fixed letter; vowel sign; suffix and post-suffix. See Figure 1 below.

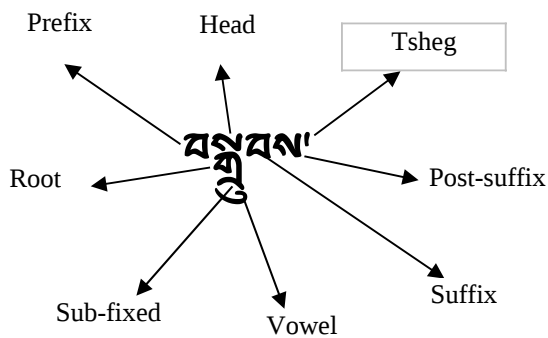


Figure 1. Elements of a Dzongkha Syllable

a) Root letter: The most vital letter in a syllable since it defines the sound of the syllable, it is the starting point for the sound of the syllable. Hence it is always pronounced first. Therefore identifying this letter is very important (<http://www.leartibetan.com>).

A letter in a syllable is always a root letter if:

1) It has a vowel,

e.g,

ཚོམ་	ch oe s	-pronounced as ' ch oe 0 ' -root letter is ' ch ' with vowel ' oe '
------	----------------	---

2) Or, it has a head letter (superscribed) or sub-fixed (subscribed) letter,

བསྐྱེད་	s ch oe s	-pronounced as ' ch oe 0 ' -root letter is ' ch ' with vowel ' oe '
ལྷམོ་	zh b m o	-pronounced as ' zh a m 0 ' -root letter is ' zh ' with sub-fixed letter ' b '

3) Or, in a two letter syllable with no explicit vowel, root letter is the first letter,

e.g,

ཤར་	sh r	-pronounced as ' sh a r 0 ' -root letter is the first letter ' sh '
-----	-------------	--

4) Or, in a three letter syllable with no explicit vowel, root letter is usually the middle letter unless the last letter is the post-suffix letter. In which case sometimes the root letter is the first letter, or it is the second letter,

e.g,

རངས་	r ng s	-pronounced as ' r a ng 0 ' -root letter is the first letter ' r '
མདུས་	m d s	-pronounced as ' d e 0 ' -root letter is the second letter ' d ' with post-suffix ' s '

5) Or, in a four letter syllable with no explicit vowel, the second letter is the root letter.

e.g,

གསརས་	g s r p	-pronounced as ' s a p 0 ' -root letter is ' s '
-------	----------------	---

b) Head-letters: Only three letters can be the head-letters. They are **r, l** and **s**.

1) The Ra-go letters:

ཀ་ ཁ་ ར་ ལ་ ཅ་ ཉ་ ཏ་ ཐ་ ད་ ན་ པ་ བ་ མ་ ཙ་ འ་

-**'r'** can be allowed as head letter for only twelve letters; k, g, ng, j, ny, t, d, b, m, ts and dz.

-They are all pronounced the same as the root letters.

e.g, ཀ་ is pronounced as **'k a 0'**.

2) The La-go letters:

ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་

-**'l'** can be allowed as a head letter for only ten letters; k, g, ng, c, j, t, p, b & h

-All except one are pronounced the same as the root letters. That is, ཀླ་ is pronounced as **'l hh a 0'**.

3) The Sa-go letters:

སྐ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་ སྒ་

-**'s'** can be allowed as a head letter for only eleven letters; k, g, ng, ny, t, n, p, b, m and ts

-Most pronounced same as the root letters except for སྐ་ སྒ་ and སྒ་ , in this case each of the root letter's pronunciation is modified to that of a high tone.

c) Sub-fixed (subscribed) letters: Four letters; y, r, l and t are sub-fixed with few root letters as follows:

i) The Ya-ta letters: here the letter **'y'** is the subscribed letter: ཡ་ ར་ ལ་ ཅ་ ཉ་ ཏ་ ཐ་ ད་ ན་ པ་ བ་ མ་ ཙ་ འ་

ཡ་ and ར་ are pronounced as **'c a 0'**

ལ་ and ཅ་ are pronounced as **'ch a 0'**

ཉ་ and ཏ་ are pronounced as **'j a 0'**

ii) The Ra-ta letters: Here the letter **'r'** is the subscribed letter.

ཀ་ ཁ་ ག་ ན་ པ་ བ་ མ་ ཙ་ འ་ ཡ་ ར་ ལ་ ཅ་ ཉ་ ཏ་ ཐ་ ད་ ན་ པ་ བ་ མ་ ཙ་ འ་

ཀ་ ར་ ལ་	tr a 0
ཁ་ ར་ ལ་	thr a 0
ག་ ར་ ལ་	dr a 0
ན་	nr a 0
པ་	mr a 0
བ་	shr a 0
མ་	sr a 0
ཙ་	hr a 0

Table 8: Pronunciation of the Ra-ta letters

Note: The ra-ta letters usually form clusters.

iii)The La-ta letters: Here the letter **'l'** is the subscribed letter:

ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་ ཀླ་ ཀྲ་ ཀྱ་

-All are pronounced as **'l a 1'** except for ཀླ་ ' which is pronounced as **'d a 0'**

iv) Fifteen Wa-zur letters: Here the letter **'b'** is the subscribed letter: ཡ་ ར་ ལ་ ཅ་ ཉ་ ཏ་ ཐ་ ད་ ན་ པ་ བ་ མ་ ཙ་ འ་ ཡ་ ར་ ལ་ ཅ་ ཉ་ ཏ་ ཐ་ ད་ ན་ པ་ བ་ མ་ ཙ་ འ་

-All are pronounced the same as the root letters.

d) Prefixes: These are unpronounced letters in the beginning of a syllable before the root letter. Though they are never pronounced, some of them modify the pronunciation of some root letters. There are five prefixes; g, d, b, m and h or ག་ ད་ བ་ མ་ ཏ་

མཚོ་	m tsh o	-pronounced as 'ts o 0' -m is silent (this is because the root letter here is ts)
------	----------------	---

དབང་	d b ng	-pronounced as 'w a ng 1' -Here the root letter b 's pronunciation is modified to that of w 's and also the tone of the whole syllable is modified.
------	---------------	--

e) Suffixes: These are the letters written after the root letters. Suffixes 'hh' and 's' are never pronounced. Pronunciation of the remaining suffixes depends on the combining root letters. That is, depending on the combining root letters those suffixes may or may not add their own sounds. There are ten suffixes; g, ng, d, n, b, m, hh, r, l and s or ག ས ཉ ཏ ལ མ འ ར ལ ས

འཛིག་	hh j i g	-pronounced as 'j i g 0' -suffix g is pronounced along with root letter j
བཀག་	b k g	-pronounced as 'k a 0' -here suffix g is not at all pronounced

Note: As seen earlier, some suffixes are capable of modifying the vowel sound in a syllable. And all suffixes can combine with all thirty initial consonants (root consonants) to produce a word or a syllable.

f) Secondary-suffixes: These are letters that follow the suffix letters. They can also be called as post-suffixes. The letters 's' and 'd' are the post-suffixes for the language. They are not pronounced at all in a syllable.

ལྷགས་	l c g s	- pronounced as 'c a g 0'
-------	----------------	---------------------------

Note: Post-suffix 'd' is used rarely. When used it can combine with suffixes 'n', 'r', & 'l' only, whereas the frequently used post-suffix 's' can combine with only four suffixes, 'g', 'ng', 'p' and 'm' in a syllable.

4. Conclusion and Future Work

This paper presented the phonemic analysis for Dzongkha language. It also covered the basic pronunciation rules for the language. With the idea that the sound system for the language is guided by certain pronunciation rules, it can be noted that arrangement of letters in a syllable or word greatly influences the pronunciation of that particular syllable or word. An extensive research on the language is required to discuss other possibilities that may influence the pronunciation of words.

There are possibilities that the language may have more than two tones as indicated by van Driem and Tshering (1998), which presents the opportunity for researchers to gather information and compare data using phonemic and acoustic analysis, to be certain of their existence. That is going to be one of our future endeavours.

Acknowledgement

This research study has been supported by the PAN Localization Project (www.panl10n.net) grant. We would like to thank the PAN regional secretariat and the team from Human Language Technology (HLT) of National Electronics and Computer Technology Centre (NECTEC) in Thailand for their technical guidance and continuous support.

References

George van Driem and Karma Tshering, (Collab). 1998. Languages of Greater Himalayan Region.

George van Driem. The Grammar of Dzongkha. 1992. Dzongkha Development Commission. Thimphu, Bhutan.

Uden Sherpa, Dawa Pemo, Dechen Chhoeden, Anocha Rugchatjaroen, Ausdang Thangthai, and Chai Wutiw WATCHAI. 2008. Pioneering Dzongkha text-to-speech synthesis. *Proceedings of the Oriental COCOSA*, 150–154. Kyoto, Japan.

The New Dzongkha Grammar. 1999. Text Book. Dzongkha Development Commission, Thimphu, Bhutan.

Grapheme-to-Phoneme Conversion for Amharic Text-to-Speech System

Tadesse Anberbir

Ajou University, Graduate School of
Information and Communication,
South Korea.

tadesse@ajou.ac.kr

Michael Gasser

Indiana University, School
of Informatics and Computing,
USA.

gasser@indiana.edu

Tomio Takara

University of the Ryukyus,
Graduate School of Engineering
and Science, Japan.

Kim Dong Yoon

Ajou University, Graduate School of
Information and Communication,
South Korea.

Abstract

Developing correct Grapheme-to-Phoneme (GTP) conversion method is a central problem in text-to-speech synthesis. Particularly, deriving phonological features which are not shown in orthography is challenging. In the Amharic language, geminates and epenthetic vowels are very crucial for proper pronunciation but neither is shown in orthography. This paper describes an architecture, a preprocessing morphological analyzer integrated into an Amharic Text to Speech (AmhTTS) System, to convert Amharic Unicode text into phonemic specification of pronunciation. The study mainly focused on disambiguating gemination and vowel epenthesis which are the significant problems in developing Amharic TTS system. The evaluation test on 666 words shows that the analyzer assigns geminates correctly (100%). Our approach is suitable for languages like Amharic with rich morphology and can be customized to other languages.

1 Introduction

Grapheme-to-Phoneme (GTP) conversion is a process which converts a target word from its written form (grapheme) to its pronunciation form (phoneme). Language technologies such as Text-to-speech (TTS) synthesis require a good GTP conversion method.

GTP conversion comes under two main approaches: rule-based and data-driven techniques and recently some statistical techniques have been proposed (See (Damper et al., 1998) for review of the several techniques). Using these methods successful results are obtained for different languages (Taylor, 2005; Chalamandaris et al., 2005) and other. However, in many languages automatic derivation of correct pronunciation

from the grapheme form of a text is still challenging. Particularly phonological features which are not shown in orthography make the GTP conversion very complex.

Amharic, the official language of Ethiopia, has a complex morphology and some phonological features are not shown in orthography. Morphology, the way morphemes in a language join to form words, influences language technology because some phonological processes cannot be modeled without proper modeling of morphological processes. For example, most geminates in Amharic language are related to grammatical processes and can be predicted from morphological processes. In general, for Semitic languages such as Amharic and Tigrinya, morphological analysis can make explicit some of the phonological features of the languages that are not reflected in the orthography and plays a very crucial role in text-to-speech synthesis. However, so far no study has been conducted in this area.

In this study, we proposed and integrated a preprocessing morphological analyzer into an Amharic Text-to-speech (AmhTTS) system mainly to automatically predict geminates and epenthetic vowels positions in a word. Our research is the first attempt to integrate a morphological analyzer called HornMorpho (Gasser, 2011) into an AmhTTS. The integrated morphological analyzer takes Amharic Unicode input and outputs Latin transcription marking geminates and the location of epenthetic vowels. Then, the output of the morphological analyzer is used by the AmhTTS system and further processed to extract all the features generated by the analyzer. AmhTTS system is a parametric and rule-based system designed based on cepstral method (Anberbir and Takara, 2006).

The paper is organized as follows: Section 2 provides background information about Amharic language and Section 3 briefly discusses about Amharic writing system and challenges in GTP conversion. Then, Section 4 and Section 5 describe about the automatic assignment method using morphological analyzer and evaluation results, respectively. Finally, Section 6 presents concluding remarks.

2 The Amharic language

Amharic, the official language of Ethiopia, is a Semitic language that has the greatest number of speakers after Arabic. According to the 1998 census, Amharic has 17.4 million speaker as a mother tongue language and 5.1 million speakers as a second language (Ethnologue, 2004).

A set of 34 phones, seven vowels and 27 consonants, makes up the complete inventory of sounds for the Amharic language (Baye, 2008). Consonants are generally classified as stops, fricatives, nasals, liquids, and semi-vowels. Table 1 shows the phonetic representation of the consonants of Amharic as to their manner of articulation, voicing, and place of articulation.

Table 1. Categories of Amharic Consonants with corresponding IPA representation.

Manner of Articulation	voicing	Place of Articulation					
		Labials	Dentals	Palatals	Velars	Labio-velar	Glottals
Stops	Voiceless	ፕ [p]	ቲ [t]	ቸ [tʃ]	ከ [k]	ኩ [k]	አ [ʔ]
	Voiced	ብ [b]	ድ [d]	ጅ [dʒ]	ግ [g]	ግ* [g]	
	Glottalized	ቶ [pʰ]	ጥ [tʰ]	ጭ [tʃʰ]	ቅ [q]	ቅ* [qʰ]	
Fricatives	Voiceless	ፍ [f]	ሰ [s]	ሸ [ʃ]			ህ [h]
	Voiced	ቨ [v]	ዘ [z]	ሻ [ʒ]			
	Glottalized		ጽ [sʰ]				
	Rounded						ሁ* [hʷ]
Nasals	Voiced	ጠ [m]	ን [n]	ሻ [ɲ]			
Liquids	Voiced		ለ [l]				
			ረ [r]				
Glides	Voiced	ው [w]			ይ [j]		

The seven vowels, along with their representation in Ge'ez characters, are shown in terms of their place of articulation in Table 2. In addition to the five vowels common among many languages, Amharic has two central vowels, /ə/ and /i/, the latter with a mainly epenthetic function. The epenthetic vowel /i/ plays a key role in syllabification. Moreover, in our study we found the epenthetic vowel to be crucial for proper pronunciation in Amharic speech synthesis.

Table 2. Categories of Amharic Vowels with IPA equivalent.

	front	central	back
High	አ [i]	አ [i]	አ [u]
Mid	ኤ [e]	አ [ə]	አ [o]
low		አ [a]	

Like other languages, Amharic also has its own typical phonological and morphological features that characterize it. The following are some of the striking features of Amharic phonology that gives the language its characteristic sound when one hears it spoken: the weak indeterminate stress; the presence of glottalic, palatal, and labialized consonants; the frequent gemination of consonants and central vowels; and the use of the automatic epenthetic vowel (Bender et al., 1976). Among these, we found gemination of consonants and the use of the automatic epenthetic vowel to be very critical for naturalness in Amharic speech synthesis.

3 Amharic writing system and Problems in GTP Conversion

Amharic uses the Ge'ez (or Ethiopic) writing system which originated with the ancient Ge'ez language. In this system the symbols are consonant-based but also contain an obligatory vowel marking. Most symbols represent consonant-vowel combinations, but there may also be a special symbol for each consonant that represents the plain consonant. Each Amharic consonant is associated with seven characters (referred to as "orders") for the seven vowels of the language. It is the sixth-order character that is the special symbol, representing the plain consonant. The basic pattern for each consonant is shown in Fig. 1, where: C=Consonant and [] shows vowels in IPA.

1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
C[ə]	C[u]	C[i]	C[a]	C[e]	C	C[o]
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ

Figure 1. Amharic syllabic structure with example for consonant ል /l/.

Amharic writing system is partially phonetic. According to (wolf 1995), there is more or less a one-to-one correspondence between the sounds and the graphemes. However, as shown in Table 3, it has some features that make it complex from the perspective of GTP conversion.

In what follows we discuss the two main ambiguities in Amharic orthography in more detail.

Table 3. Problems in Amharic grapheme-to-phoneme (G2P) conversion.

Problem	Example
Homograph	Depending on the context, the word ገፍ can have the meaning 'still/yet' or 'Christmas'
Insertion of epenthetic vowel [i]	in words like ትምህርት, epenthetic vowel should be inserted and pronounced as /t i mhirt/ not /tmhrt/
Introduction of semi-vowel w, y	words like, በቅሎአቸን bəqlo-afin 'our mule' becomes በቅሎዋቸን bəqlowafin.
Compression of successive vowels	ለ + ኣኔ /lə + ine/ becomes ለኔ /ləne/ የአማርኛ yə-amari na becomes yamari na

3.1 Gemination

Gemination in Amharic is one of the most distinctive characteristics of the cadence of the speech, and also carries a very heavy semantic and syntactic functional weight. Unlike English language in which the rhythm of the speech is mainly characterized by stress (loudness), rhythm in Amharic is mainly marked by longer and shorter syllables depending on gemination of consonants, and by certain features of phrasing (Bender and Fulass, 1978). In Amharic, all consonants except /h/ and /ʔ/ may occur in either a geminated or a non-geminated form. Amharic, and other languages written with the Ge'ez script, differs from most other languages that feature gemination, such as Japanese, Arabic, Italian, and Tamil, in that gemination is not shown in the orthography.

Table 4. Minimal pairs with Singleton vs. Geminate Consonants

Singleton			geminate	
Orth.	Pronunc.	Gloss	Pronunc.	Gloss
ገፍ	gəna	still/yet	gənnna	Christmas
ለጋ	ləga	fresh	ləgga	he hit
ሰፊ	səfi	tailor	səffi	wide
ሸፍታ	ʃifta	outlaw	ʃiffita	rash
ይሰማል	yisəmal	he hears	yissəmmal	he/it is heard

Amharic gemination is either lexical or morphological. As a lexical feature it usually cannot be predicted. For instance, ገፍ may be read as /gəna/, meaning 'still/yet', or /gənnna/, meaning 'Christmas'. (See Table 4 for some similar example of minimal pairs). Although this is not a problem for Amharic speakers because minimal pairs are relatively infrequent, it is a challenging problem in Amharic speech synthesis. In fact, the failure of the orthography of Amharic to show geminates is the main challenge in GTP conversion that we found in our research. Without a

context, it is impossible to disambiguate such forms.

On the other hand, when gemination is morphological, rather than lexical, it is often possible to predict it from the orthography of the word alone. This is especially true for verbs (Bender and Fulass, 1978). For example, consider two words derived from the verb root consisting of the consonant sequence sbr 'break', ሰበረው and ይሰበራሉ. The first is unambiguously /sibərəw/ 'break (masc.sing.) it!', the second unambiguously /yissəbbəralu/ 'they are broken'. The fact that the /s/ and /b/ are not geminated in the first word and are both geminated in the second and that the /r/ is geminated in neither word is inferable from the prefix, the suffix, and the pattern of stem vowels. That is, within the verb there is normally some redundancy. Therefore, with knowledge of the lexical and morphological properties of the language, it is possible to predict gemination.

3.2 Epenthesis

Epenthesis is the process of inserting a vowel to break up consonant clusters. Epenthesis, unlike gemination is not contrastive and it is not surprising that it is not indicated in the orthography of Amharic and other languages. But, although it carries no meaning, the Amharic epenthetic vowel /i/ (in Amharic 'ሰርጎ ገብ' Baye, 2008) plays a key role for proper pronunciation of speech and in syllabification. However, the nature and function of the epenthetic vowel has been a problem in Amharic studies and so far no study has been conducted on the phonetic nature of this vowel.

As noted above, Amharic script does not distinguish between consonants that are not followed by a vowel and consonants that are followed by the high central vowel /i/, and as shown in Fig.1, both are represented by the sixth order (ሳድስ) character in a series. The sixth order characters are ambiguous; depending on their position in a word; they can be either voweled (with epenthic vowel /i/) or unvoweled. For example, in ልብ /libb/ 'heart', the first character, ል, represents the CV sequence /li/(voweled), whereas in ስልክ /silkk/ 'telephone', the same character represents the bare consonant /l/(unvoweled). Because such differences are crucial for speech synthesis, a TTS system needs access to the epenthesis rules.

4 Proposed Amharic GTP Conversion Method

In this section, first we discuss about Amharic morphology and the ‘HornMorpho’ morphological analyzer which we integrated into the AmhTTS system. Then, we briefly discuss our proposed GTP conversion method.

4.1 Amharic Verb Morphology

Amharic is a morphologically complex language. As in other Semitic languages such as Arabic, Amharic verbs consist of a stem – analyzable as a combination of a lexical root and a pattern representing tense, aspect, mood, and various derivational categories – and various affixes representing inflectional categories (Bender and Fulass, 1978). Verb roots consist of consonant sequences, most of the three consonants. The patterns that combine with roots to form verb stems consist of vowels that are inserted between the consonants and gemination of particular root consonants.

Consider the role of gemination in distinguishing the three main categories of three-consonant (triradical) roots, traditionally referred to as types A, B, and C (Bender and Fulass, 1978). Table 5 shows the pattern of gemination and the positions of vowels for triradical roots in the five basic tense-aspect-mood categories of the language. In type A, except in the perfective form, there is no gemination at all. Type B is characterized by the gemination of the second consonant throughout the conjugation. Type C differs from the other two types in that the second consonant is geminated in the perfective and imperfective aspects only (Amsalu and Demeke, 2006). The system is complicated by the fact that each root can also occur in combination with up to 10 derivational categories, such as passive and causative.

Table 5. Vowel-gemination patterns for triradical roots (adopted from Amsalu and Demeke, 2006)

Verb Types	Type A	Type B	Type C
Perfective	C1VC2C2C3-	C1VC2C2VC3	C1VC2C2VC3-
Imperfective	-C1VC2C3(-)	-C1VC2C2C3(-)	-C1VC2C2C3(-)
Imperative	C1C2VC3(-)	C1VC2C2C3(-)	C1VC2C3(-)
Gerund	C1VC2C3-	C1VC2C2C3-	C1VC2C3-
Infinitive	-C1C2VC3	-C1VC2C2C3	-C1VC2C3

The patterns shown in Table 5 are for the simple derivational category, which can be seen as the default for most roots, but each cell in the table could be augmented with up to 9 other patterns. These different derivational categories also

affect gemination. For example, for the passive category, the imperfective pattern for Type A becomes -C1C1VC2C2VC3(-), with both first and second consonants geminated.

Based on the gemination patterns, a morphological analyzer can be used to locate geminates in an input word form. For example, the Amharic word ይሰበራል ‘it is broken’ is correctly pronounced with gemination (lengthening) of the first and second consonants: yissəbbəral. The gemination in this case is grammatical, and a morphological analyzer can infer it based on its knowledge of Amharic verb roots and the particular patterns that they occur with.

4.2 ‘HornMorpho’ Morphological Analyzer

Amharic morphology, especially verb morphology, is extremely complex (Baye, 2008), but it is relatively well understood. Thus it is possible, with considerable effort, to create a morphological analyzer for the language using finite state techniques. HornMorpho (Gasser, 2011) is such a system. Given a word in conventional Amharic orthography, it infers the gemination of consonants in the word wherever this is possible (as well as extracting grammatical and lexical information from the word). The rules for epenthesis in Amharic are also quite complex and not completely understood, but a first approximation has been implemented in HornMorpho. That is, the output of HornMorpho includes the epenthetic vowel /i/ wherever it is expected, according to the rules in the program, effectively disambiguating the sixth-order orthographic characters and simplifying the overall GTP conversion.

When analyzing, the word is first Romanized using a variant of the SERA romanization system. Next the program checks to determine whether the word is stored in a list of unanalyzable or pre-analyzed words. If not, it attempts to perform a morphological analysis of the word. It first does this using a “lexical” approach, based on a built-in lexicon of roots or stems and its knowledge of Amharic morphology. If this fails, it tries to guess what the structure of the word is, using only its knowledge of morphology and the general structure of roots or stems. If the “guesser” analyzer fails, the program gives up. Both the lexical and guesser analyzers operate within the general framework known as finite state morphology¹.

¹ For a more in-depth introduction to finite state morphology, see Beesley & Karttunen(2003), for another application of

4.3 Finite State Morphology

Finite state morphology makes use of transducers that relate surface forms with lexical forms. In the case of HornMorpho, a surface form is the orthographic representation of an Amharic word, for example, ያስበራል *yisəbəral*, and the corresponding lexical form is the root of the word and a set of grammatical features. For ያስበራል, the lexical representation is *sbr* + [sbj=3sm, asp=imprf, vc=ps], that is, the root *sbr* ‘break’, third person singular masculine subject, imperfective aspect, and passive voice, in English, ‘he/it is broken’.

Each of the arcs joining the states in a finite state transducer (FST) represents an association of an input character and an output character. Successful traversal of an FST results in the transduction of a string of input characters into a string of output characters. In HornMorpho, using a technique developed by Amtrup (2003), we add grammatical features to the arcs as well, and the result of traversing the FST is the unification of the features on the arcs as well as the output string. This allows us to input and output grammatical features as well as character strings.

A simple FST can represent a phonological or orthographic rule, a morphotactic rule (representing possible sequences of morphemes), or the form of a particular root. Because gemination is both lexical and grammatical in Amharic, it plays a role in all three types. By combining a set of such FSTs using concatenation and composition, we can produce a single FST that embodies all of the rules necessary for handling words in the language. Since FSTs are reversible, this FST can be used in either direction, analysis (surface to lexical forms) or generation (lexical to surface forms). HornMorpho uses a single large FST to handle the analysis and generation of all verbs. This FST includes a lexicon of verb roots as well as all of the hundreds of phonological, orthographic, and morphological rules that characterize Amharic verbs.

4.4 Proposed GTP conversion Method

For Amharic text-to-speech, we use the existing HornMorpho analysis FST and a modified version of the HornMorpho generation FST. We first run the analysis FST on the input orthographic form. For example, for the word ያስበራል,

this yields *sbr* + [sbj=3sm, asp=imprf, vc=ps]. Next we run a phonological generation FST on this output, yielding the phonological output *yissəbbəral* for this example. The second and third consonants are geminated by an FST that is responsible for the passive imperfective stem form of three-consonant roots.

Figure 2 shows the architecture our proposed method. First the morphological analyzer accepts Unicode text from file and generates the corresponding phonemic transcription with appropriate prosodic markers such as geminates and indicates the location of epenthetic vowel. Then, the output of the analyzer will be an input for AmhTTS system and further annotated by the text analysis module to extract syllables and prosodic marks. Finally the speech synthesis part generates speech sounds.

AmhTTS synthesis system is a parametric and rule based system designed based on the general speech synthesis system (Takara and Kochi, 2000). The text analysis in AmhTTS system extracts the linguistic and prosodic information from the output of a morphological analyzer and extracts the gemination and other marks and then converts into a sequence of syllables using the syllabification rule. The syllabification rule uses the following syllables structure (V, VC, V'C, VCC, CV, CVC and CV'C and CVCC).

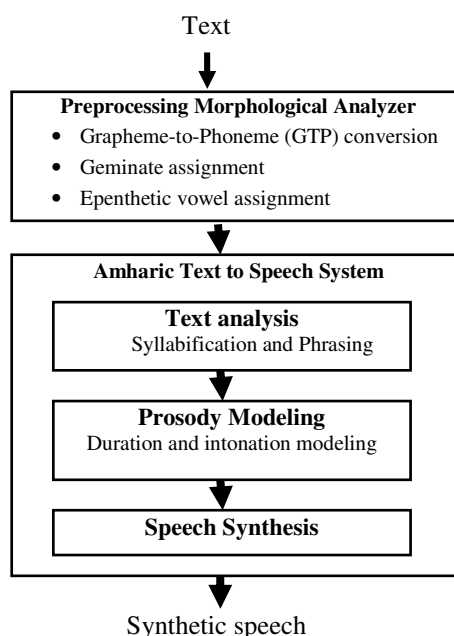


Figure 2. Proposed architecture for GTP conversion and automatic geminate and epenthetic vowel assignment

finite state technology to Amharic morphology, see Saba & Girma (2006).

5 Evaluation

A preliminary evaluation of the proposed automatic geminate assignment method was made by analyzing 666 words and we found 100% correct assignment/restoration of gemination. The words were selected from Armbruster (1908) verb collections, where gemination is marked manually, representing all of the verb conjugation classes (Type A, B and C). In Type A penultimate consonant geminates in Perfect only, in Type B penultimate consonant geminates throughout the conjugation and in Type C penultimate consonant geminates in Perfect and Contingent.

However, the analyzer does not analyze words that contain non-Ge'ez characters or Ge'ez numerals, and also there is an incomplete list of common words (with gemination) which it does not attempt to analyze. For example, given the unambiguous form ይሰበራሉ, it outputs /yisbebberallu/, and given the ambiguous form ገፍ, it outputs both possible pronunciations /gəna/ and gənnə/. Words like ገፍ can only be inferred by analyzing the context and finding out the parts-of-speech (POS) of the word. But this is beyond the scope of the current work.

6 Conclusion

The paper discussed orthographic problems in Amharic writing system and presented preliminary results on a method for automatic assignment of geminates and epenthetic vowel in GTP conversion for Amharic TTS system. Our method is the first attempt for Amharic language and can be easily customized for other Ethio-Semitic languages.

However, the work described in this paper is still on progress and for more accurate GTP conversion, parts-of-speech (POS) tagger and phrase break predictor needs to be implemented or addressed. For example, the word ገፍ, which can be pronounced as both / gəna/ meaning 'still/yet' and gənnə meaning 'Christmas', can only be inferred by analyzing the context and finding out the POS of the word.

In the future, we have a plan to disambiguate words like ገፍ using syntax and use the analyzer to extract more features and finally customize our TTS system for other languages.

Acknowledgments

The authors would like to thank Daniel Yacob for providing the Armbruster verb collections where the gemination is marked manually with the "Tebek" symbol (U+135F).

References

- A. Chalamandaris, S. Raptis., and P. Tsiakoulis. 2005. *Rule-based grapheme-to-phoneme method for the Greek*. In: Proc. of INTERSPEECH-2005, pp. 2937-2940, Lisbon, Portugal.
- Amtrup, J. 2003. *Morphology in machine translation systems: efficient integration of finite state transducers and feature structure descriptions*. Machine Translation, 18, 213-235.
- Armbruster, C. H. 1908. *Initia Amharica: an introduction to spoken Amharic*, Cambridge: Cambridge University Press.
- Baye Yimam. 2008. የ አ ማር ኛ ሰ ዋ ሰ ው (Amharic Grammar), Addis Ababa. (in Amharic).
- Beesley, K.R., and Karttunen, L. 2003. *Finite state morphology*. Stanford, California: CSLI Publications.
- Ethnologue. 2004: *Languages of the World*, <http://www.ethnologue.com/>
- Gasser, M. (2011). HornMorpho: a system for morphological analysis and generation of Amharic, Oromo, and Tigrinya words. *Conference on Human Language Technology for Development*, Alexandria, Egypt.
- Leslau, Wolf. 1995. *Reference Grammar of Amharic*, Wiesbaden: Harrassowitz.
- M.L Bender, J.D.Bowen, R.L. Cooper and C.A. Ferguson. 1976. *Language in Ethiopia*, London, Oxford University Press.
- M. Lionel Bender and Hailu Fulass. 1978. *Amharic Verb Morphology: A Generative Approach*, Carbondale.
- Paul Taylor, 2005. *Hidden Markov Model for Grapheme to Phoneme Conversion*. In: Proc. of INTERSPEECH-2005, pp. 1973-1976.
- R.I. Damper, Y. Marchand, M. J. Adamson, and K. Gustafson. 1998. *A comparison of letter-to-sound conversion techniques for English text-to-speech synthesis*", Proceedings of the Institute of Acoustics, 20 (6). pp. 245-254.
- Saba Amsalu and Girma A. Demeke. 2006. *Non-concatinative Finite-State Morphotactics of Amharic Simple Verbs*, ELRC Working Papers Vol. 2; number 3.
- T. Anberbir and T. Takara. 2006. *Amharic Speech Synthesis Using Cepstral Method with Stress Generation Rule*, INTERSPEECH 2006 ICSLP, Pittsburgh, Pennsylvania, pp. 1340-1343.
- T. Takara and T. Kochi. 2000. *General speech synthesis system for Japanese Ryukyu dialect*, Proc. of the 7th WestPRAC, pp. 173-176.

Design of a Text Markup System for Yorùbá Text-to-Speech Synthesis Applications

Odétúnjí Àjàdí , ODÉJOBÍ
Computer Science & Engineering Department
Faculty of Technology
Obáfémí Awólówò University
Ilé-Iḡè, Nigeria
oodejobi@oauife.edu.ng

ABSTRACT

The text intended as input into a text-to-speech (TTS) synthesis system can come from a number of digital sources, including: electronic emails, text-books, Internet webpages and newspapers. In order to generate the speech sound corresponding to the input text, a TTS system must extract the information relevant for computing speech signal and associated prosody. The ease with which this information can be extracted depends on a number of factors, including: the orthography of the target language, the domain of the text, and whether the content of the text is constrained by some writing rules or standards. Usually the information that accompanies such text are inadequate for computing the prosody of the corresponding text. A text markup system is then used to provide the missing information. This paper discusses how this task is achieved for the Standard Yorùbá language.

Keywords

text, spech synthesis, markup, text tagging

1. INTRODUCTION

The orthography of Standard Yorùbá (SY) has been described as *shallow* (Bird, 1998), implying that it is relatively easy to generate pronunciation from the text without the need for complicated syntactic analysis; such as part-of-speech analysis. For example, the pronunciations of the English words *record*(verb) and *record* (noun) differ because of their syntactic classes. This type of situation does not occur in SY. In the (SY) language orthography, diacritic marks are used to represent tones. This tonal information removes the ambiguity in the pronunciation of a well written and properly accented SY text. The additional information provided by tonal marks is very important in the computation of intonation and prosody for a piece of text in Text to Speech Synthesis (TTS) system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HLTD 2011 Alexandria, Egypt
Copyright 2011 HLTD – ...\$10.00.

Despite the phonetic nature (i. e. closeness of the orthography to pronunciation) of the SY orthography, the information in the text is not enough for computing an accurate prosody in text to speech application (Odéjobí et al., 2008). The orthography of SY, and indeed any other language, is designed to provide enough information for a human reader. Paralinguistic aspects of expressions, such as emotion - which endows speech with its naturalness quality - cannot be reduced into writing but are normally guessed by human readers based on context. Other information, such as the structure and style for pronouncing the text are not explicitly represented in the orthography. There is, therefore, the need to augment the content of the text with additional information that will facilitate an unambiguous processing of prosodic data in an input text. In this paper we shall discuss the the design of a text markup system for adding additional prosody information into a SY text intended as input to a Text to Speech Synthesis (TTS) system.

1.1 The nature of written text

A written text can be described as an encoding of a speech utterance using symbols. The text encodes two important information: (i) the content and manner of speech, and (ii) how it should be spoken. The content of the speech, i.e. its written form, is defined by the letters, symbols and numerals in the text. The function of how an entity will be spoken is specified by punctuation marks such as comma (,), full-stop (.), semicolon (;), exclamation mark (!), question mark (?), etc. A space or a hyphen can also be used to indicates entities that are to be pronounced separately or as a discrete unit. For example, a space between two words indicates that each word is a single entity to be pronounced separately; whereas a hyphen used in the same context indicates that the words should be pronounced as if they are a single entity.

The domain of text considered in this work is derived from Yorùbá newspapers and standard textbooks. This class of text has two important attributes: (i) a *physical content*, and (ii) a *logical structure*. The physical content is described by tokens which form the component of the text. This includes the following:

- punctuation marks, including spaces;
- lexical items written using the SY orthography;
- numerals;
- symbols and acronyms;

- lexical items written using a foreign orthography (such as English words and proper names).

At a higher level is the grammar that guides the construction and organisation of the symbols and letters into syllables, words, phrases, and sentences. The logical structure of a text specifies how the text is organised into pronunciation units. The following elements constitute the logical structure of an SY text of reasonable length: (i) a title, (ii) one or more paragraphs, (iii) sentences, (iv) phrases, (v) words, (vi) syllables. The logical structure of an SY document can be described declaratively, and without reference to how a formatting program should realise the desired format.

1.2 Text models

There are three types of text model that can result from the above described SY text structure, namely: (i) *restricted*, (ii) *unrestricted*, and (iii) *markup*. In the restricted text model, specific rules that guide the acceptable contents and structure of the text are explicitly defined. For example, a rule may specify that all numerals or abbreviations in a text must be written in their lexical format. Another rule may specify the type and acceptable location of punctuation marks that must be used in the text. All input text must be written according to the defined rules and hence the input text is forced to conform to a format. The problem with restricted input is that it makes the TTS machine more difficult to use since the input format rules are normally not standardised and are difficult to standardise since there are divergent writing styles and genre of the text, e.g. drama and poem. In addition the text may be written to convey different concepts which will be compromised if the content or style of the text is restricted.

A softcopy of unconstrained text can be represented and stored in a plain text format (e.g. using UNICODE or ASCII). This type of representation is, however, not powerful enough for describing special features of the text that can provide a meaningful input for TTS machines. For example, in plain text, although there are several popular methods, there is no universally agreed convention for delimiting paragraph boundaries.

In the unrestricted text model, there is no restriction on the input text. The text may contain any symbol, letter or number and can represent text from diverse domains such as weather forecast, poems, incantation, etc. The text-preprocessing module of the TTS machine is then required to extract information necessary for synthesising the text. This approach imposes a more complicated text analysis task on the TTS machine. Predicting the prosodic attributes of the text by using automatic or rule based techniques is unlikely to produce an acceptable level of accuracy. (Quené and Kager, 1992) argued that this task can only be accurately done if the automatic process has access to the semantic and pragmatic information about the input text.

In a markup text model, the input text, usually unrestricted, is annotated with information that renders the prosody of the intended utterance transparent. Markup input can easily be built around a standard markup language, such as the eXtensible Markup Language (XML) making them easy to use by large group of users. Among the information that can be included in an annotated text include intonation phrases, phonetic attributes as well as descriptions of how foreign words and other text anomalies should be pronounced.

Many markup scheme for TTS systems, have been proposed particularly for non-tone languages (Taylor and Isard, 1997, Ogden et al., 2000). But such markup schemes are system specific and often use annotation schemes not specifically tailored for tone languages texts.

1.3 Issues in SY typesetting and markup

A very important feature of a fully tone-marked SY text is that the tones and under-dots are adequately indicated, hence tonal information of each syllable can be extracted from the text. This type of text can be typesetted using \LaTeX . The standard \LaTeX fonts have all of the components required to compose a Yorùbá text (Taylor, 2000). This is because the \LaTeX provides markup for indicating diacritic and under-dots associated on letters. This features makes it possible to conveniently generate text with the correct SY orthography. The tones and phones are the two most important ingredients for generating accurate SY pronunciation.

Another feature of SY orthography, which the \LaTeX system also represents accurately, is the use of unambiguous spaces between words. This information can be used to determine word boundaries. Therefore the typesetting of SY texts in \LaTeX ensures that accented character format can be used to accurately represent the input to SY TTS. \LaTeX provides annotation at the lower level of the physical content of text thereby facilitating the incorporation of more accurate utterance information. In addition, \LaTeX also facilitates the generation of portable documents, such as PDF and .DVI files, which can be read by humans and efficiently stored and transmitted.

On the word level, however, information about how a word must be pronounced in different contexts, e.g. rate of speech, speaking style, etc., cannot be adequately specified using \LaTeX . Besides, the logical structure of text which controls the overall prosody of an utterance is better defined at levels high than word (Quené and Kager, 1992). Also, the same sequence of words can be read in different manners or styles depending on the context. Phrases, sentences, and paragraphs are not explicitly specified in \LaTeX except through the use of punctuation marks, such as full-stop, comma, and semi-colon, and other escape sequences like the carriage return. For example, in the case of sentences, full-stop normally used for delimiting declarative statements may be ambiguous in some situations, e.g. in sentences also containing numbers with fractions, for example 12.45.

Predicting the phrase and sentence boundaries is a complicated problem if a \LaTeX typesetted text is to be processed directly by a TTS system. Moreover, in some speech synthesis tasks, such as text containing dialogue, it may be required to control specific characteristics of the generated speech such as the loudness, rate of speech, age, and gender of the speaker, etc. A TTS system requires phrase and sentence level information to generate the appropriate prosody for a given text. For example, the information on whether a sentence is a question or an exclamation is best defined at sentence and phrase level since they affect utterance units longer than words.

Since \LaTeX is so effective in describing the physical content of the text, a reasonable approach would be to design another markup system above \LaTeX , which will describe the logical structure of the text. This will allow us to effectively describe the more abstract higher level prosody of the text.

2. TEXT MARK-UP IN SPEECH SYNTHESIS

When a piece of text is read aloud, how much information from the text is responsible for the sound-waveform generated? The text certainly conveys clues to the grouping of syllables in word sequences. It also contains cues about important syntactic boundaries that can be identified by way of punctuation marks. SY text, if written with the complete orthographic specification, contains cues about tones and syllables as well as their organisation into words, phrases and sentences. The identification and appropriate use of these cues can greatly improve the computation of an accurate intonation for the text.

In general, however, such syntactic cues are not enough to facilitate the generation of adequate prosody from text. For example, the same sequence of words can be spoken in different ways, depending on the context and mode. What really matters is the marking of structure and contents in such a way that pauses and emphases are placed correctly and the hierarchy of phrasing and prominence is equitably conveyed (Monaghan, 2001). (Taylor and Isard, 1997) have observed that plain text is the most desirable form of input to a TTS system from a human perspective due to its standard nature and universal understanding. Therefore, there is the need to render input text in such a way that it can be easily read by a human as well as easily processed by a TTS system.

The best way to achieve this goal is to explicitly annotate the input text with information that will aid further processing of the text. The idea of a text markup language was first introduced by (Goldfarb et al., 1970) with the design of the Generalised Markup Language (GML), which later evolved into the Standard Generalised Markup Language (SGML). Since then, a number of text markup languages have been developed and used.

Many TTS developers have designed text Markup Languages (ML) specifically for their TTS applications (e.g. (Taylor and Isard, 1997, Huckvale, 1999, Huckvale, 2001)). Some of these ML include: Spoken Text Markup Language (STML) and Speech Synthesis Markup Language (SSML) (Taylor and Isard, 1997, Burnett et al., 2002), VoiceXML (VoiceXML, 2000) as well as the Java Speech Markup Language (JSML) (JavaSpeechML, 1997). JSML was developed to facilitate a standard text markup and programming for TTS engine in the Java environment. Most of these mark-up languages provide text description tags that describe the structure of the document, and speaker directive tags that control the emphasis, pitch rate, and pronunciation of the text.

SABLE (Sproat et al., 1998) is a TTS markup language developed by combining STML and two other markup languages, i.e. Java Speech Markup Language (JSML) and Speech Synthesis Markup Language (SSML), to form a single standard. It is designed to be easily implemented in any TTS engine. It has been implemented in both the Edinburgh University's Festival speech synthesis system and the Bell Labs TTS engine (Taylor and Isard, 1997).

The following is a simple example of SABLE markup for SY two sentence paragraph: "Bàbá àgbè ti ta cocoa 30 kg ní N500 kí ó tó mò pé àjọ NCB ti fi owólé cocoa. Ni n'kan bíi dédè agogo 3:00 òsán ni bàbá àgbè délé" (meaning "[Father framer has sold cocoa 30 kg before he knows that organisation NCB has add money to cocoa. At about time 3:00 afternoon fa-

ther farmer got home.] The farmer has sold 30 kg of cocoa for N500 before realising that the NCB organisation has increased cocoa price. The farmer got home around 3:00 in the afternoon"):

```
<DIV TYPE="paragraph">
  <DIV TYPE="sentence" >
    Baba agbe ti ta cocoa 30 kg ni
    N500 ki o to mo pe ajo
    NCB ti fi owole cocoa.
  </DIV>
  <DIV TYPE="sentence">
    Ni n'kan bii dede agogo
    3:00 osan ni baba agbe dele.
  </DIV>
</DIV>
```

In this example, the structure of the text to be pronounced is clearly marked. SABLE also includes tags to specify numeral and other text anomalies.

The markup systems discussed above are not suitable for SY text because they do not adequately describe data and structures found in typical SY text. Using them for marking SY text will lead to a complex representation system which is difficult to use. An alternative suggested by (Huckvale, 2001) is to provide an open, non-proprietary textual representation of the data structures at every level and stage of processing. In this way, additional or alternative components may be easily added even if they are encoded in different format. This approach was used in the *ProSynth* project (Ogden et al., 2000).

In *ProSynth*, each composed utterance comprising a single intonation phrase is stored in a hierarchy. Syllables are cross-linked to the word nodes using linking attributes. This allows for phonetic interpretation rules to be sensitive to grammatical function of a word as well as to the position of the syllable in the word. Knowledge for phonetic interpretation is expressed in a declarative form that operates on prosodic structures. A special language called *ProXML* was used to represent knowledge which is expressed as unordered rules and it operates solely by manipulating the attribute on XML-encoded phonological structure.

The *ProXML* implementation suggests that the facility provided by XML matches the requirements to represent the phonological features of an utterance in a metrical prosodic structure, namely: nodes described by attribute-value pairs forming strict hierarchies (Huckvale, 1999). An important attribute of this structure for prosody modelling is that the phonetic descriptions and timings can be used to select speech unit and expresses their durations and pitch contour for output with a TTS system.

The apparent success of XML at representing phonological knowledge, as well as the additional advantage that the represented text can be published on the Internet, motivated our use of XML in developing the markup system for the SY language.

3. TEXT-TO-SPEECH MARKUP SYSTEM

The review presented in the previous section reveals two important facts. First, the design of a markup system is greatly influenced by the method selected for the implementation of the TTS high level synthesis module, the features of the target language (e.g. orthography) and the degree of freedom required in the control of prosody in the TTS system.

Second, the markup scheme used in most systems are based on the eXtensible Markup Language (XML). This is partly because XML allow users to add additional control commands in a flexible and easy to implement manner. XML provides a more abstract and powerful mean of describing speech prosody at utterance level higher than the syllable (i.e. word, phrase, sentence and paragraph) and facilitates possible publications and data sharing on the Internet.

3.1 Design of XML system

The logical structure of any SY text *document* to be synthesised can be viewed as a tree. The root of the tree is associated with the entire document. The *title* of the document is an optional attribute of the root element. The first sub-tree element is the *paragraph*. A document may contain one or more paragraph elements. Each paragraph is equivalent to a branch from the root document. The second sub-tree element is the *sentence*. A paragraph may contain one or more sentences. The third sub-tree element is the *phrase* and the fifth sub-tree element is the *word*. The phrase element is made up of one or more words and each word element is made up of one or more syllables. Each syllable can be adequately typesetted using L^AT_EX by indicating to diacritic marks and under-dot as required. For example, in syllable *bó*, the diacritic mark *´* indicates the tone (i.e. high) and *bó* is the base. The vowel in the base is the letter *o* with an under-dot. Other L^AT_EX specifications for typesetting of SY text are shown in Table 1.

The structure of the XML tree defined above a L^AT_EX document is shown in Figure 1. The leaf node of the XML tree forms the root of the L^AT_EX part of the tree representation for a piece of text. Element at the same level of the tree are on same level in the linguistic structure of the utterance corresponding to the text. For example, two sentences in the same paragraph of a text share the same branch at the paragraph level. They are also on the same linguistic level in the utterance prosodic hierarchy. The text structure ends with syllable as the leaf elements.

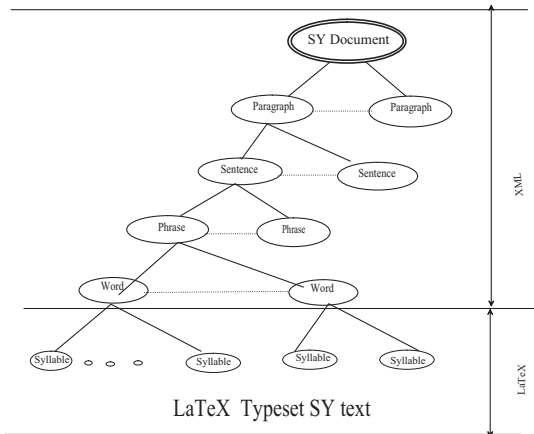


Figure 1: XML tree defined above L^AT_EX

Note that the tree is built from a hierarchical pattern of objects, each of which has a specific attribute which contributes to how the sentence is to be pronounced. We now discuss the design of higher level prosodic markup system using XML.

Table 1: L^AT_EX annotation for SY diacritic and under dots

Tag	Description	Result	Example
\’	High tone	é	Adé (Crown)
\d	Under dot	ẹ	Ọlẹ (Fetus)
\’	Low tone	è	Òlẹ (Lazy)
\ (default)	Mid tone	ē or (e)	ewé (Leaf)

4. DOCUMENT TYPE DEFINITION (DTD)

A document type definition (DTD) is a set of declarations that conforms to a particular markup syntax and that describes a class, of “type” of XML documents, in terms of constraints on the logical structure of those document (Wikipedia, 2004). In a DTD for marking text for prosody modelling, the structure of a class of documents is described via:

element definition tags representing each element required to describe the prosody of SY text, and

attribute definition the combination of rules for elements and a list of the information which can be contained within the element. These rules are used to specify the contents of each element.

The beginning of an element is specified by the *start-tag* associated with the tag name and ends with a corresponding *end-tag*. The attribute list for an element contains information which will be included in its tag. An element may have a number of attributes, each of which will have values which can be set when the element is specified. Elements that do not appear in a particular tag may be given a default values

In the design of the tag names for our XML system, we observed that the some annotations L^AT_EX can be confused with standard XML tags. The annotation name specifications in L^AT_EX has the form `\tag_name` while XML tags is of the form `<tag_name>` or `</tag_name>`. But in situation where these names can be confused, we retain the name for L^AT_EX and use the first four upper-case letters of the name for defining the XML tag. Each element is defined by a **start-tag** (e.g. `<document>`) and an end tag (e.g. `</document>`).

There are two important criteria in the design of tag names in the XML our markup system. The first is that computer-literate non-speech-synthesis experts should be able to understand and use the markup. The second is that the markup should be portable across platforms so that a variety of speech synthesis systems should be able to use the additional information provided by the tags (Taylor and Isard, 1997). The general syntax of an XML document is dictated by a set of rules defined by the World Wide Web Consortium (W3C) (Burnett et al., 2002). It consists of a grammar-based set of production rules derived from the Extended Backus-Naur Form (EBNF). In the following, we discuss and illustrate the design of each tagging system using the following SY text.

Title = Ìdàmún àgbè

Bàbá àgbè ti ta cocoa 30 kg ní N500 kí ó tó mò pé àjò NCB ti fi fowó lé cocoa. Ní ìkan bìi dédé agogo 3:00 òsán ni bàbá àgbè délé. Kíá tí àwon alágbèse tí ó bá bàbá ṣiṣe ní oko cocoa rè gbópé óti dé láti ojà ni wón wátò sí enu ònà ilé bàbá àgbè làti gba owó iṣe tí bàbá jẹ wón. Léyìn igbà tí bàbá ṣan owó àwon alágbàse tán ló tó ri wípé kò sí èrè rárá lóri oun tí òun tà. Diè ni owó tí ó kù fi lé ní N102.

4.1 The document tag

The `<document>` tag delimits the root element of an SY document. It has an optional *title* attribute which indicates the title of the document. The contents of *title* attribute include an optional text which specifies the title of the document. The `<Document>` tag encloses the text to be spoken. The syntax of the `<Document>` tag is as follows:

```
<document title="text">
[
<!-- The content of the document will go here -->
[
</document>
```

The plus sign (+) indicates that a document can contain one or more paragraphs. For example, the `<document>` tag is represented as follows:

```
<document title="Ìdàmún Àgbè" >
{
<!-- The content of the document will go here -->
}
</document>
```

4.2 The paragraph tag

The paragraph tag, `<PARA>`, defines each paragraph in the document. We use the tag name `<PARA>` so that it will not be confused with the *paragraph* annotation in L^AT_EX. The paragraph contains an optional attribute which indicates the *style* that will be used in uttering the paragraph. The syntax of the `<PARA>` tag is as follows;

```
<PARA style = "Styletype">
{
sentence+ <!(i.e. The sentences in paragraph) >
}
</PARA>
```

The style attribute accepts five style types:

1. *Dialogue*- a paragraph spoken in the context of a dialogue conversation.
2. *Spontaneous*- a paragraph spoken spontaneously.
3. *Read* - a paragraph spoken as read speech (Default).

4. *Poem* - a paragraph spoken as read Yorùbá poem, e.g. Ewì (common poem), Oríkì (praise song), etc.
5. *Incantation*- a paragraph spoken as SY incantation, e.g. Ọfọ, Ọgèdè, Àṣẹ, etc..

The default value for style is *Read*. Using the paragraph tag, the first paragraph in the sample text will be tagged as follows:

```
<document title = Ìdàmún Àgbè >
<PARA style= "Read">
<!-- The content of each paragraph will go
here -->
</PARA>
</document>
```

4.3 The sentence tag

The sentence tag, `<sentence>`, delimits an SY sentence and contained in the paragraph element. All sentence elements within the same paragraph have same paragraph attributes. A sentence element contains at least one phrase element. The sentence element has many attributes which specify the prosodic information of a sentence. These attributes are useful in a multi-speaker text, such as a play or a dialogue. It is also required for annotating texts containing more than one reading styles, e.g. a poem. The sentence attributes include the following:

MODE: specifies the mode for speaking a sentence. The mode attribute can take one of the following values: *Question, Declaration, Exclamation, Statement*. The default value is *Statement*.

MOOD: specifies the mood for the speaking a sentence. The mood attribute can assume one of the following values: *Happy, Sad, Normal*. The default value is *Normal*.

RATE: specifies the rate at which a sentence is will be spoken. The attribute has three possible values: *Fast, Normal, Slow*. The default value is *Normal*.

PAUSE: signifies the duration of the pause that should be inserted between two words using a linguistic value. This attribute has 3 possible values: *Short, Long, Medium*.

LOUD: signifies the loudness or volume of the sentence as linguistic values. The values possible: *Low, Medium, High*. The default value is *medium*.

GENDER: specifies the type of voice, i.e. male or female, for synthesising the sentence. It has the following values: *Male, Female*. The default value is *male*.

AGE: specifies the approximate age of voice to be synthesises: It has the following values: *Child, Adult, and Old*. The default value is *Adult*.

The syntax for sentence tagging is therefore:

```
<sentence MODE="Statement" MOOD="Normal"
STYLE="Oro"
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
  <!-- The content of each sentence go here -->
</sentence>
```

All the parameters specified in this syntax are the default values of the attributes.

The annotation for the first sentence in our example text is as follows:

```
<sentence MODE="Statement" MOOD="Normal"
STYLE="Oro"
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
Bàbá àgbè ti ta cocoa 30 kg ní N500 kí ó tó mò pé àjọ
NCB ti fi fowó lé cocoa.
</sentence>
```

The last two attributes specifies the kind of speaker's voice to be imitated by the speech synthesiser. If it is not specified an adult male native speaker of SY is assumed. This attribute is only useful for selecting the relevant database in a multi-speaker TTS environment, such as dialogue or story telling. The attribute will guide the TTS engine in selecting the appropriate database.

The sentence tags and attributes discussed above are designed in the manner stated above in order to facilitate the synthesis of text representing a dialogue between two or more people. This situation is very common in plays and newspaper interview texts. The scope of the speaker tag varies. In situation where only one language database is available all specified speaker attributes will be ignored.

4.4 The phrase tag

The phrase tag, <phrase>, can be inserted within the text of a sentence to indicate the presence of phrase boundaries. The <phrase> effectively breaks a sentence into intonation phrases even when punctuation marks are present. The syntax for the phrase element is as follows:

```
<phrase>
  <!-- The content of each phrase go here -->
</phrase>
```

The <phrase> tag for specifying the prosodic phrase for the previous example sentence is as follow:

```
<sentence MODE="Statement" MOOD="Normal"
STYLE="Oro"
  RATE="Normal" PAUSE="Medium" LOUD="Medium" >
<phrase> Bàbá àgbè ti ta cocoa 30 kg ní N500 </phrase>
<phrase> kí ó tó mò pé àjọ NCB ti fi fowó lé cocoa
</phrase>.
</sentence>
```

5. TEXT CONTENTS DESCRIPTION TAGS

In a normal SY text, there are many textual anomalies which can occur as part of the content. Some examples of textual anomaly include numerals representing ordinal or cardinal data items, letters representing abbreviations, as well as a groups of letters in foreign -words and proper

names, e.g. David. In order to remove the complexities involved in determining the exact pronunciation of these textual anomalies, we defined markup tags for them. These tags are built around the SAYAS tag in W3C (Burnett et al., 2002) and extended to incorporate features specific to SY text. The information provided by this tag will allow the High Level Synthesis module to determine the type of expansion to apply on each of the tagged items during text normalisation process.

5.1 The SAYAS tag

The text content elements are used to markup the content of a text in order to specify how they are to be spoken. For example, N500 must be spoken as currency; IADS is to be spoken as an acronym with specific pronunciation, whereas the abbreviation O.A.U. is to be spelt out as English alphabet using Yorùbá accent. The list of tags for content element markup is specified in Table 2.

Following the W3C format, we use the SAYAS tag with three specific attributes. The SUB attribute is used to specify a substitute text for an abbreviation or acronym. For example, the text *Kòànùn* can be substituted for COAN (COmputer Association of Nigeria). The CLASS attribute is used to specify the class of the pronunciation as stated in Table 2. When this parameter is not specified, the default is SY abbreviation pronounced with SY accent. Some examples of SAYAS tags is as follows:

```
<SAYAS SUB ="a.b.b.l."> àti bèè bèè lọ </ SAYAS>
<SAYAS SUB ="i.n.p."> ìyẹn ní pé </ SAYAS>
<SAYAS SUB ="f.w." >fiwé </SAYAS>
<SAYAS SUB ="b.a." > bí àpẹ̀rẹ̀ </SAYAS>
<SAYAS SUB ="f.a." > fún àpẹ̀rẹ̀ </SAYAS>
<SAYAS SUB = "w.o.r" > wò ó ore </SAYAS>
<SAYAS SUB = "e.n.p." > èyí nipé</SAYAS >
```

5.1.1 The SUB attribute

The syntax for the SUB attribute is

```
< SAYAS SUB = "text to be substituted" > text </SAYAS>
```

The SUB attribute is particularly useful in defining replacement text for abbreviations and other shorthand text. The substitution strings for some commonly used SY abbreviation are defined as bellow:

5.1.2 The CLASS attribute

The syntax for the CLASS attribute is as follows:

```
<SAYAS CLASS = "attribute" > text </SAYAS>
<SAYAS CLASS ="currency" > N500</SAYAS>
<SAYAS CLASS ="acronym" > AIDS </SAYAS\>
```

5.1.3 The ABBRACENT attribute

The third attribute is the abbreviation accent attribute, ABBRACENT. It determines whether an abbreviation will be spelt out as Yorùbá abbreviation using a Yorùbá accent or an English abbreviation (each letter is from the English

alphabet) using Yorùbá accent. For example, the abbreviation “a.b.b.l.” (i.e. àti bèè bèè lọ) is a Yorùbá abbreviation and its component alphabet must be pronounced using SY phones. However, O.A.U. (Organisation of African Unit) is an English abbreviation which must be pronounced using SY accent. Below is an example of the usage of the above markup tags:

```
<SAYAS CLASS =”ABBREVIATION”
ABBRACENT=’English’ > O.A.U </SAYAS>
```

6. CONCLUSIONS

The text markup system for text intended as input to standard Yorùbá speech synthesis system is presented. Future work will be directed at the text normalisation system that will use the information provided in the text to expand textual anomalies.

Acknowledgments

Financial support from the *African Language Technologies Initiative* and encouragement from Dr. Tùndé Adégbolá and Professor Kólá Owólábi to the development of the ideas presented in this paper is acknowledged and highly appreciated. The support from the Ọbáfẹ̀mí Awólówò University is here acknowledge.

7. REFERENCES

- Bird, S. (1998). Strategies for representing tone in African writing systems: a critical review. URL:<http://cogprints.org/2174/00/wl2.pdf>. Visited: Jan 2003.
- Burnett, D. C., Walker, M. R., and Hunt, A. (2002). Speech synthesis markup language version 1.0 w3c working draft 02. <http://www.w3.org/TR/speech-synthesis/#S1.1>. visited: Jun 2004.
- Qdėjóbi, O. A., S., W. S. H., and Beaumont, A. J. (2008). A modular holistic approach to prosody modelling for standard yorùbá speech synthesis. *Computer Speech & Language*, 22:39–68.
- Goldfarb, C. F., Mosher, E. J., and Peterson, T. I. (1970). An online system for integrated text processing. In *Proc. American Society for Information Science*, volume 7, pages 147–150.
- Huckvale, M. (1999). Representation and processing of linguistic structures for an all-prosodic synthesis system using xml. In *Proc. of EuroSpeech ’99*, number 4, pages 1847–1850, Budapest.
- Huckvale, M. (2001). The use and potential of extensible markup (XML) in speech generation. In Keder, E., Bailly, G., Monaghan, A., Terken, J., and Huckvale, M., editors, *Improvements in Speech Synthesis: Cost 258: The Naturalness of Synthetic Speech*, chapter 30, pages 298–305. Wiley Inter. Science.
- JavaSpeechML (1997). Java speech markup language (jsml) specification , version 0.5. Visited: May 2005.
- Monaghan, A. (2001). Markup for speech synthesis: a review and some suggestions. In Keder, E., Bailly, G., Monaghan, A., Terken, J., and Huckvale, M., editors, *Improvements in Speech Synthesis: Cost 258: The Naturalness of Synthetic speech*, chapter 31, pages 307–319. Wiley Inter. Science.

- Ogden, R., Hawkins, S., House, J., Huckvale, M., Local, J., Carter, P., Dankovicova, J., and Heid, S. (2000). Prosynth: an integrated prosodic approach to device-independent natural-sounding speech synthesis. *Computer Speech & Language*, 14:177–210.
- Quené, H. and Kager, R. (1992). The derivation of prosody for text-to-speech from prosodic sentence structure. *Computer Speech & Language*, 6:77–98.
- Sproat, R., Hunt, A., Ostendorf, M., Taylor, P., Black, A., Lenzo, K., and Edgington, M. (1998). Sable: A standard for TTS markup. <http://www.bell-labs.com/project/tts/sabpap/sabpap.html>. Visited: Jun 2004.
- Taylor, C. (2000). Typesetting african languages. http://www.ideography.co.uk/library/afrolin_gua.html. Visited: Apr 2004.
- Taylor, P. and Isard, A. (1997). SSML: A speech synthesis markup language. *Speech Communication*, 21:123–133.
- VoiceXML (2000). Voice extensible markup language: VoiceXML. <http://www.voicexml.org/specs/VoiceXML-100.pdf>. Visited: May 2003.
- Wikipedia (2004). Document type definition. http://en.wikipedia.org/wiki/Document_Type_Definition. visited: Jul 2005.

Appendix

Table 2: Tags for SY text

Token Classification	Description	Tag name
Date	String of numbers formatted as 99-99-99, or 99/99/99	DATE
Time	String of numbers formatted as 99:99, 99/99,	TIME
Currency	String of numbers prefixed by a currency symbol, e.g. N, \$, £,	CURRENCY
Lexical	String of letters	LEXICAL
Ordinal digits	String of numbers prefixed by a noun	ORDINAL
Cardinal digit	String of numbers postfixed by a noun	CARDINAL
Loanword	Word with foreign language spelling, e.g. English, French, Arabic, e.t.c.	LOAN
Punctuation	Punctuation marks such as (;) , (:), (.)	PUNCT
Acronym	Group of upper case letters such as FIFA, OAU, USA, e.t.c.	ACRONYM
Special Character	Characters such as *, +, etc.	SPEC
SI unit	SI unit to be expanded into SY accent pronunciation	SUNIT
Phone	Phone number (digit by digit pronunciation)	PHONE
Proper names	Proper names, usually of English origin	PRONAME

Comparing Two Developmental Applications of Speech Technology

Aditi Sharma Grover¹

¹CSIR Meraka Institute,
P.O. Box 395,
Pretoria, 0001,
South Africa

asharmal@csir.co.za

Etienne Barnard^{1,2}

²Multilingual Speech Technologies
Group, North-West University,
Vanderbijlpark, 1900
South Africa

etienne.barnard@gmail.com

Abstract

Over the past decade applications of speech technologies for development (ST4D) have shown much potential for enabling information access and service delivery. In this paper we review two deployed ST4D services and posit a set of dimensions that pose a conceptual space for the design, development and implementation of ST4D applications. We also reflect on these dimensions based on our experiences with the above-mentioned services and some other well-known projects.

1 Introduction

The use of speech technology for developmental purposes (ST4D) has seen slow but steady growth during the past decade (for an overview, see Patel, 2010). However, much of that growth has been opportunistic in nature: researchers or development agencies determine an environment where speech technology (ST) may potentially be useful, and then deploy a variety of methods and tools in order to deliver useful services (and ask appropriate research questions) in that environment. To date, no overarching framework that could assist in this process of service identification – design – implementation – analysis has emerged.

Having been involved in a number of such efforts ourselves, we have experienced the lack of theoretical guidance on the application of ST4D as a significant challenge. Hence, we here attempt some initial steps in that direction. In particular, we review two services that we have deployed, sketch a number of important ways in which they differed from one another, and use that comparison to derive some abstract dimensions that partially span the space of ST4D applications. Of course, these are early days for ST4D, and much remains to be discovered. We nevertheless believe that the explication of this

space will be useful guidance for further research in this field.

2 Two deployed ST4D services

We describe in this section two telephone-based services that were designed for developing world regions with varying contexts and goals, and use them in subsequent sections to illustrate the various dimensions related to design and deployment of ST4D.

2.1 Lwazi

The Lwazi Community Communication Service (LCCS) is a multilingual automated telephone-based information service. The service acts as a communication and dissemination tool that enables managers at local community centres known as Thusong community centres (TSCs) to broadcast information (e.g. health, employment, social grants) to community development workers (CDWs) and the communities they serve. The LCCS allows the recipients to obtain up-to-date, relevant information in a timely and efficient manner, overcoming the obstacles of transportation, time and costs incurred in trying to physically obtain information from the TSCs. At a high-level, the LCCS can be viewed as consisting of 3 distinct parts as illustrated in Figure 1.

During our investigations we found that TSCs often need to communicate announcements to the CDWs, who in turn disseminate the information to the relevant communities. The TSC managers and local municipal (government) communication officers most often have regular Internet connectivity and use emails in their daily routine to communicate with government departments and non-profit organisations (NPOs).

In the majority of the communities we investigated, communication with the CDWs is mostly through face-to-face meetings and the telephone. CDWs on average had grade 8-12 or higher level of education (vocational certificate courses),

ranged between 20-45 years in age, with no significant differences in the gender proportions. They were familiar with mobile phone usage and some even used computers and the Internet. Almost none of the community members have access to the Internet, while most of the households have access to at least a mobile phone.

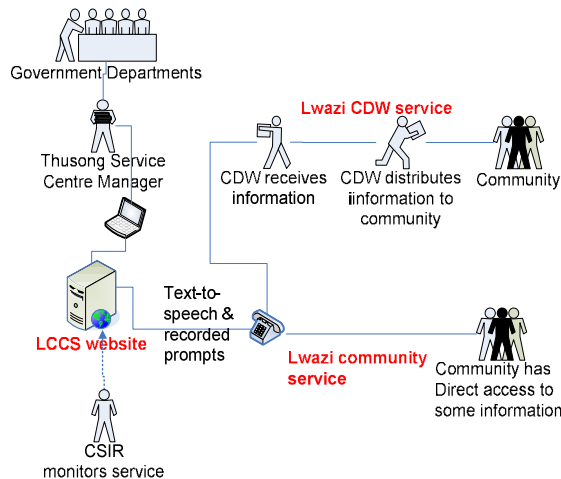


Figure 1. Overview of the LCCS.

2.1.1 LCCS website

This site provides a TSC manager with the ability to upload new announcements to the service for CDWs and/or community members. Managers may choose to send an announcement to all the CDWs registered in their area, or select group, and/or the rest of the community. The manager provides information on relevant fields (date, time, etc.) and selects the languages he/she would like the message to appear in. For each additional language selected, additional text-boxes are provided to type the message text for those languages. The audio structure of the announcements as heard by the community workers is shown in figure 2. A 'View Messages' page also provides managers with a voicemail-like facility to listen to messages (new and old) on the LCCS website left by CDWs through the Lwazi CDW service.

2.1.2 LCCS telephone services

Two IVR services, namely the Lwazi CDW service and the Lwazi community service were developed. The former allows CDWs to phone in and get access to the announcements uploaded by the TSC managers and leave voice messages for the TSC managers. Each CDW in a targeted area is registered on the CDW service and ob-

tains a PIN code that allows them to access their announcements.

Within each targeted area the service was provided in the most commonly spoken languages of that area. Daily at 5 pm, all the recipients receive an SMS notification if they have new message(s) on the LCCS. The service is free in the sense that a CDW only gives a missed call to the service (calls the service number and hangs up after a ring), and the service then calls them back. The CDW also gets the option in the main menu to record a voice message for the TSC manager. This feature was envisaged to save the CDW the cost of a phone call to leave a voice message for the TSC manager using their mobile phone (i.e. to directly dial the TSC manager's number vs. using the LCCS). The TSC manager, in turn, receives all her/his community workers' related messages through the LCCS website where he/she can easily store and retrieve them as required.

The announcement is played back using a combination of pre-recorded voice prompts for the dates and time fields and TTS (in blue) for the less predictable fields like the message text.

```

Lwazi (user is called back on registered number): Welcome to the Lwazi Service. You have 2 new messages. To hear your New Message, press 1. To hear your Old Messages, press 2. To Leave a Message for the Thusong centre manager, press 3. Note that you can return to the 'Start' of the service at any time, by pressing 0.
User: (presses 1)
Lwazi: New Messages. Note that you can skip to the next message, by pressing 1. First message:
Lwazi: Description of Meeting: <(TTS:) Weekly progress report meeting>
Date: <29th> of <March>
Venue: <(TTS:) Thusong Service centre>
Starting Time: <3> <30> <PM>
End time: <5> <PM>
Message: <(TTS:) All CDWs are requested to attend this weekly meeting where community participation strategies will be discussed>.

```

Figure 2. Lwazi CDW service- sample script.

The Lwazi community service (second IVR) extends from the Lwazi CDW service; if a TSC manager wishes to make announcements to the community in general, community members are marked as recipients on the LCCS website. Therefore, the same announcement which goes out to the CDWs is then accessible to the community members as well. Community members similarly give the service a missed call, which calls them back making the service free for their usage. However, the community line differs from the CDW line as no registration is required by the communities (i.e. no PIN codes thus any user

can call in or SMS notifications of a new message on the service).

The LCCS was piloted at 6 sites in South Africa across the 11 official South African languages. The pilot areas included rural (3), semi-rural (2), and urban areas (1). Note, these pilots were intended as short-term deployments running on average 4-12 weeks to determine the uptake and usage trends in each area.

2.2 OpenPhone

The OpenPhone service is a general health information line designed for caregivers of HIV positive children in Botswana. A caregiver is any individual who takes care of an HIV positive child, e.g. parents, family or community member. The vast majority of caregivers are females and range between the ages of 18 and 65 with most being semi and low literate. Most caregivers tend to have low-income jobs and many are often unemployed. Baylor, a HIV paediatrics hospital where children receive free treatment provides the caregivers with free lectures on various aspects of living with HIV and caregiving. Each caregiver on average attends two lecture sessions. It was observed that caregivers often forget the material covered in the lectures. Reinforcement through written material is not viable as many caregivers are semi-to-low literate. The Baylor lectures and all interactions with caregivers are in Setswana (local language) since most caregivers are uncomfortable with English. Baylor staff explains complex health information in accessible terms in the local language. Mobile phone usage and ownership was widely prevalent (up to 90%).

It was also observed that caregivers often travel large distances (average 28 km and as far as 500 km) with high costs and time spent during a working day. Also they often have general health information queries (e.g. nutritional needs, hygiene, etc.) beyond the material covered in the lectures and although caregivers are encouraged to call Baylor with any questions they may have, most are reluctant (and unable) due to the high costs of mobile phone calls. These challenges and issues formed the basis of the design for OpenPhone, an IVR health information service in Setswana and accessible at any time through a simple telephone call. A sample system-user interaction is shown in figure 3.

Openphone was piloted for a week at Baylor where the service was tested with 33 caregivers with one of the major questions being the preference of input modality between automatic speech

recognition (ASR) and touchtone (DTMF), where ASR was simulated by using the wizard of Oz (WoZ) methodology (Fraser & Gilbert 1991). Two identical systems were built that differed only at the menu prompts in choice of input modality, e.g. a DTMF menu option would be; “to hear about Nutrition, press 1,” whereas the ASR menu option would say, “to hear about Nutrition, say Nutrition.”

On average caregivers reported that mobile phone costs per month were 68 Pulas (\$10.5 USD) with an average cost per call being reported as 4.5 Pulas (\$0.75 USD). Only 30% of caregivers reported having access to a landline telephone and of these only 9% had the landline at home.

System (Introduction): *Hello and Welcome to the Health Helpline, I am Nurse Lerato and I know you probably have many questions about caring for someone with HIV.*
System (Overview): *I can tell you about Hygiene & Cleanliness, Nutrition, Common Sicknesses, ARV Medication, and Facts about HIV. If at any time during your call you want to start our conversation again, you can press 0.*
System (Main Menu): *For Hygiene & Cleanliness, please press 1, for Nutrition, press 2, for Common Sicknesses, press 3, for ARV medication, press 4 or for Facts about HIV, please press 5.*
User: [Presses 2.]
System: *Eating a balanced diet of different foods helps people with HIV stay healthy and strong. A healthy diet does not have to be costly and contains food from all the different food groups. Healthy food is always prepared and stored in a clean environment...*

Figure 3. OpenPhone – sample script.

3 Charting the ST4D space

In this section we posit a set of dimensions that pose a conceptual space for the design, development and implementation of ST4D. We reflect on these dimensions based on our experiences with the above-mentioned projects and some other well-known projects.

3.1 Nature of the user community

Various user-related factors are vital in ST4D. These include literacy, the technology experience of the user and the ‘openness’ of the user community.

Undoubtedly *literacy* of the target users is one of the major considerations in any information and communications technology for development (ICTD) project, and speech applications have often been proposed to address user interface problems with other modalities experienced by low and semi-literate users (Plauché et al, 2007; Medhi et al, 2007). However, Barnard et al (2008) highlighted that, without training or prior exposure, a large class of speech applications

may not be usable for the developing world user. In OpenPhone for example, at the end of one call, a user proceeded to ask the ‘nurse’ (system persona) a question when prompted by the system to leave a comment (and waited for the answer), highlighting that some users within low and semi-literate populations’ may not fully realise that a service is automated.

Technology experience coupled with literacy may also affect an application’s uptake. The uptake of mobile phones in the developing world is widespread (Heeks, 2009); users who are comfortable with related features of mobile technologies (e.g. SMS, USSD, a service provider’s airtime loading system) may perform better on speech applications. In OpenPhone we found that experience ‘loading airtime’¹ correlated positively with higher task completion. We also found that loading airtime was the sole significant factor in user preference of DTMF over the ASR system i.e. those people who loaded airtime regularly preferred DTMF over ASR. These experiences perhaps indicate that prior technological experience may be just as important a factor in adopting new technology as literacy.

Another factor with great variegation is the ‘openness’ of the user community. Here we refer to how membership to the user community is determined; e.g. in Lwazi the user community was essentially closed: a set of CDWs who were paid government workers. This leads to a set of repeat users, who are also trainable if required. In contrast, in OpenPhone the user community was more open in that any caregiver (or anyone with the application’s phone number) could call the application. However, it would be hard to identify repeat users (except through caller ID) and even harder to provide such a user community with training.

This factor also affects design choices; in catering for a closed community, known commonalities between users allow the designer to fine-tune the application towards their specific needs. In contrast, a more open community would have various types of users (e.g. cultural, language, age, literacy differences) where adapting the application becomes much more difficult due to the user diversity. For example, in Lwazi our pilot sites varied from rural to urban as well as across

11 languages, which made the system persona design quite a challenge.

3.2 Content source

Generation of content is a critical issue for not only ST4D, but any ICTD application. Here it is of essence to provide the users with *relevant and timely content*. The *content source* may be locally (user/community) generated or externally generated by the system designers themselves. This of course, has implications on *content updates and its timeliness*. For example, in Lwazi the content was generated within the community by the TSC manager (and sometimes CDWs) – thus, it was very locally relevant, could be easily updated and was timely in nature but this led to a large reliance on the role of the TSC managers to provide an announcements ‘feed’ to the application. In contrast, in OpenPhone the content was purely externally generated removing the reliance factor but placing a large burden in the design process to investigate and ensure that content provided was relevant, useful and timely.

Capturing high-quality information in a developing-world situation is a significant challenge in itself. The existence of electronic information sources that provide content feeds for the ST4D application (as are typically employed in the developed world) would be ideal when building such applications. However the nature of the information space in the developing world is such that these sources are generally unavailable. Thus, building them from scratch often takes bulk of the effort required in designing such speech applications. Also, the fact that the content needs to be available (or translated) in the *local language* further magnifies the issue.

It is also worthwhile to note that the *sensitivity of the content* may also affect the choice of content source. User generated content may not be feasible (and legally possible) in domains with highly sensitive information such as health as in the case of OpenPhone. Interestingly, even in other domains such as agriculture, Patel et al (2010) found that farmers preferred to obtain information from known and trusted experts rather than other framers. This goes to illustrate that content *trustworthiness* is also a significant factor.

3.3 Application complexity

In comparison to the HLT investment in the developed world, much advancement still needs to be made for the developing world resource-scarce languages. “How may I help you?” type

¹ Loading airtime refers pre-paid phones which require users to load money by calling the network provider’s service number and entering a sequence of digits from the pre-paid calling card which can be done through IVR or USSD.

of applications are certainly a long way off, rather we propose that a *'human in the loop'* approach may be required in the interim such that technology constraints and inefficiencies do not prohibit the uptake of speech-based interfaces. In our initial pilots of Lwazi the CDWs said that they sometimes struggled to hear their messages (TTS parts) and felt that voice sounded somewhat 'funny' and robotic and as if a non-mother tongue speaker was trying to speak a language.

Thus, to ensure that a TTS audio announcement was sufficiently intelligible, we introduced a 'human-in-the-loop', where every time an announcement is posted, a notification was sent to our TTS support team which checked the quality of the rendered audio file for the announcement and, if required, normalised the input text (e.g. special characters and acronyms) to render better quality TTS audio that would be posted to the telephone services. Similarly an IVR-based service could be used in the front-end of question-answering service with humans in the back-end answering questions and channelling them back to the user via TTS or SMS.

3.4 Business model and deployment

The success of ST4D applications is subject to the same provisions as many other ICTD applications. One of the major factors is *cost*; applications that require even the cost of a local phone call may be prohibitively expensive for many users in the developing world. We found in OpenPhone that the majority of caregivers said that even though the service would be useful to them they would only be able to make use of it if the service is toll-free. An average phone call in Botswana of 5-10 minutes to the service would cost a mobile phone user \$1-2 USD. The average cost per month for mobile phone usage was \$10.5 USD. Thus, a single phone call to the service would consume, 10-20% of a caregiver's monthly mobile phone budget, making the case for a toll-free number all the more imperative.

In Lwazi, a notable observation was that community members sometimes struggled to understand how the service could be free if they needed airtime (prepaid balance) in order to give it a missed call. Community members would often be wary of using their own phones to test the service, afraid that their airtime would be used. We tried to address this by showing them (with our phones) that no charge was incurred upon calling the service. In some cases, we also noted that people did not have enough airtime to even give a missed call to the service.

Based on these experiences and those highlighted by Patel *et. al* (2010) we stress that users are extremely sensitive to call costs and the need to pay for such a service, which makes cost a decisive factor in the widespread uptake of a speech application. To our knowledge all ST4D efforts reported on to date have been 'free' initiatives (i.e. the user does not pay) with the exception of the BBC Janala project (Heatwole, 2010) where users pay subsidized call rates to access English lessons over an IVR.

Stakeholder support is a related aspect of great importance; one needs to ensure that the relevant stakeholders (users, community organizations, researchers, donors, etc.) involved in the project support the application and understand the practical roles they play in the ability of the technological intervention to solve a problem. In Lwazi we found that, of the six pilot sites, the one with the highest usage was where the keenness of the TSC manager and the government communication officers was the strongest. The latter, in particular, were young, open to trying new technologies and quite familiar with the Internet. They were enthusiastic to try out the service if it would assist in making their jobs easier and also contacted us to provide valuable feedback and suggestions for improvement. Thus the role of such intermediaries cannot be over-emphasized. Clear planning must be executed around the financial, organisational, operational support and human capacity required from these stakeholders.

Related closely to the above stakeholder support factor is the need to *align new ST4D applications with existing channels*. Rather than trying to replace an existing system (resistance from the status quo) it may be better to introduce a complementary service that rather leverages on existing channels and provides an added advantage. In OpenPhone the health info line was meant as a means to augment the Baylor lectures and not replace them.

In Lwazi we found that in the pilot sites where there was a fairly workable system in place for communicating with the CDWs, the service did not fare well. For example, in one site the TSC manager just preferred to communicate with the CDWs via telephone and did not mind the cost factor as the calls were government-sponsored. In another site, CDWs had government sponsored Internet access and laptops (a very rare occurrence). Their primary means of communication with the TSC manager was through face-to-face meetings or email. The TSC seemed to be

operating quite well, and the existing channels set up for communication between CDWs and the TSC manager were actively used and worked well for their needs.

Two of the CDWs mentioned that their offices were located (unusual for CDWs to have fully-fledged offices) close to that of the TSC manager and they may therefore not be using the service so much. Another suggested that the service be linked to their emails. From this, we surmised that the service only supplemented these existing channels rather than leverage on them and thus was not so widely used.

The above discussion illustrates that a *sustainability model* is essential for a speech applications' deployment. Here, the obvious potential options to consider include government subsidies, toll-free lines sponsored by telecommunications providers (this has been surprisingly rare in developing regions) but also consider models around marketing companies that target emerging markets or user-paid services (finding a strong value proposition for the user which makes them willing to pay for the service as in the case of BBC Janala). The sustainability challenge will prove to be a significant one in the long-term deployment of such applications.

Also relating to sustainability, it is important to take cognisance of the fact that *pilot deployments may be viewed exactly as that by the community* and therefore may not obtain their full buy-in, i.e. if the community knows that a service will only be available for a short period of time, they may be less inclined to use it than if it were introduced as a permanent solution. In general, ICTD researchers need to carefully balance this aspect with that of not creating false expectations when introducing ICT interventions into communities.

4 Conclusion

All of the dimensions discussed in Section 3, create a rather complex space which leads to very different classes of applications due to choices made on the various dimensions involved. These choices affect the design of a ST4D application in several different ways, including the *input modality* (touchtone vs. speech recognition), *menu design* (hierarchical vs. non-linear), *prompts* (TTS vs. pre-recorded) and system *persona* design.

As the application of ST4D becomes more widespread, this conceptual space will undoubtedly be understood in much more detail, and the

characteristics of the most useable regions within the space will be defined with rigor and precision. We believe that such systematic knowledge will greatly enhance our ability to deliver impactful services.

Acknowledgments

This work was funded and supported by South African Department of Arts and Culture (Lwazi) and OSI/OSISA (OpenPhone). The authors wish to thank the numerous HLT research group members but especially Madelaine Plauche, Tebogo Gumede, Christiaan Kuun, Olwethu Qwabe, Bryan McAlister and Richard Carlson who played various roles in the Lwazi and OpenPhone projects.

References

- Agarwal, S., Kumar, A., Nanavati, A., and Rajput, N. 2009. Content creation and dissemination by-and-for users in rural areas. In *Proc. IEEE/ACM Int. Con. On ICTD (ICTD 09)*. April 2009, 56-65.
- Barnard, E., Plauche, M.P., Davel, M. 2008. The Utility of Spoken Dialog Systems. In *Proc. IEEE SLT 2008*, 13-16.
- Fraser, N.M., and Gilbert, G.N. 1991. Simulating speech systems, *Computer Speech and Language*, 5(2), 81-99.
- Heeks, R., 2009. IT and the World's 'Bottom Billion'. *Communications of the ACM*, vol. 52(4), 22-24.
- Medhi I., Sagar A., and Toyama K. 2007. *Text-Free User Interfaces for Illiterate and Semiliterate Users*, Information Technologies and International Development (MIT Press), 4(1), 37-50.
- Patel, N., Chittamuru, D., Jain, A., Dave, P., Parikh, T.P. 2010. Avaaj Otaalo — A Field Study of an Interactive Voice Forum for Small Farmers in Rural Indi. In *Proc. CHI 2010*, 733-742.
- Plauche M.P., Nallasamy U., Pal J., Wooters C., and Ramachandran D. 2006. Speech Recognition for Illiterate Access to Information and Technology. In *Proc. IEEE Int. Conf. on ICTD06*.
- Plauche M.P. and Nallasamy U. *Speech Interfaces for Equitable Access to Information Technology*, Information Technologies and International Development (MIT Press), vol. 4, no. 1, pp. 69-86, 2007.
- Sharma Grover, A., Plauche, M., Kuun, C., Barnard, E., 2009. HIV health information access using spoken dialogue systems: Touchtone vs. speech. In *Proc. IEEE Int. Conf. on ICTD 2009 (ICTD 09)*, 95-107.
- Sharma Grover, A., and Barnard, E., 2011, The Lwazi Community Communication Service: Design and Piloting of a Voice-based Information Service. In *Proc. WWW 2011 (to appear)*.
- Sherwani, J., Palijo, S., Mirza, S., Ahmed, T., Ali, N., and Rosenfeld, R. 2009. Speech vs. touch-tone: Telephony interfaces for information access by low literate users. In *Proc. IEEE Int. Conf. on ICTD*, 447-457.

Phonetically balanced Bangla speech corpus

S.M. Murtoza Habib¹ Firoj Alam¹ Rabia Sultana¹ Shammur Absar Chowdhur^{1,2} Mumit Khan^{1,2}

{murtoza, firojalam, ummi.ulab.bu}@gmail.com,
{shammur, mumit}@bracu.ac.bd

¹Center for Research on Bangla Language Processing, BRAC University

²Department of Computer Science and Engineering, BRAC University

Abstract

This paper describes the development of a phonetically balanced Bangla speech corpus. Construction of speech applications such as text to speech and speech recognition requires a phonetically balanced speech database in order to obtain a natural output. Here we elicited text collection procedure, text normalization, G2P¹ conversion and optimal text selection using a greedy selection method and hand pruning.

Index Terms — Phonetics, Balanced corpus, Speech Synthesis, Speech Recognition.

1 Introduction

The goal of this study is to develop a phonetically balanced Bangla speech corpus for Bangladeshi Bangla. With nearly 200 million native speakers, Bangla (exonym: Bengali) is one of the most widely spoken languages of the world (it is ranked between four² and seven³ based on the number of speakers). However, this is one of the most under-resourced languages which lack speech applications. Some sparse work has been done on speech applications of this language. General speech applications such as speech synthesis and speech recognition system require a phonetically balanced speech corpus. One of the great motivations of this work is to develop such a corpus, which will in turn help the people to develop speech synthesis and speech recognition applications. However, corpus designing is a long and tedious task and therefore optimization

is necessary, since recording every possible speech unit is not pragmatic. The “phonetic coverage” (Santen and Buchsbaum, 1997) is appropriate when we say phonetically balanced corpus. This coverage can be defined by the concept of phonetic unit. Sentence containing phonetic units according to their frequency of occurrence in a given language is called “phonetically balanced sentence” (Dafydd et al., 2000). Consequently, the corpus containing the phonetically balanced sentences is called “phonetically balanced corpus”. The phonetic units can be phone, biphone, triphone, or syllable. Many studies (Kominek and Black, 2004), (S. Kasuriya et al., 2003) and (Patcharika et al., 2002) showed that, the use of biphone as the basic unit for speech corpus is feasible during text selection process. Therefore we wanted to develop a phonetically balanced speech corpus based on biphone coverage using the widely used greedy selection algorithm (Santen and Buchsbaum, 1997) (François and Boëffard, 2002) (François and Boëffard, 2001). The study of allophones and other spoken realizations such as phonotactic constraint are beyond the scope of this paper.

The work presented here is the design, construction, and characterization of text selection procedure optimized in terms of phonetic coverage. We hope that this work will be the preliminary step to develop a speech corpus which will provide necessary resources for speech synthesis and speech recognition of Bangla. Due to the lack of this resource, experts in this field have been unable to develop speech applications for this language.

A brief literature review is given in section 2, followed by a description of the development procedure in section 3. The analytical results are presented and discussed in section 4, where we have also presented a comparison between our technique and the technique used in arctic Komi-

¹Grapheme to Phoneme

²<http://www2.ignatius.edu/faculty/turner/languages.htm>, Last accessed April, 2011.

³http://en.wikipedia.org/wiki/List_of_languages_by_total_speakers, Last accessed April, 2011.

nek and Black, (2004) database. A summary and conclusion of the study are given in section 5.

2 Literature review

It is already well established that the success of speech research mostly depends on speech corpora. Since speech corpora contain the variation of real phenomena of speech utterances, thus we are able to analyze the phenomena from speech corpus. Speech related research such as phonetic research, acoustic model and intonation model can be drawn from a speech corpus. Research on speech synthesis shows great improvement on the quality and naturalness of synthesized speech which adapted speech corpus (Kominek and Black, 2004) (Gibbon, 1997). Likewise, the development of a speech recognition system largely depends on speech corpus. The inspiration of this work came from (Kominek and Black, 2004) (Fisher et al., 1986) (Yoshida et al., 2002) (Pacharika et al., 2002) (Radová and Vopálka, 1999) where significant amount of work have done for different languages. There is a claim by LDC-IL⁴ that, a phonetically balanced Bangla speech corpus is available. Besides that, CDAC⁵ has also developed speech corpora which are publicly available through the web. There is a speech corpus publicly available for Bangladeshi Bangla - CRBLP⁶ speech corpus (Firoj et al., 2010), which is a read speech corpus. The CRBLP speech corpus contains nine categories of speech but it is not phonetically balanced. Such categories are Magazines, Novels, Legal documents (Child), History (Dhaka, Bangladesh, Language movement, 7th March), Blogs (interview), Novel (Rupaly Dip), Editorials (prothom-alo) and Constitution of Bangladesh. However, to the best of our knowledge there is no published account. In addition, due to the differences in the writing style as well as the phonetic structure between Indian and Bangladeshi Bangla we have decided to compile a phonetically balanced Bangladeshi Bangla speech corpus based on phone and bi-phone coverage, which will be the first of its kind for Bangladeshi Bangla.

⁴ <http://www.ldcil.org/>

⁵ http://www.kolkatacdac.in/html/txttospeeh/corpora/corpora_main/MainB.html

⁶ CRBLP - Center for Research on Bangla Language Processing

3 Development procedure

This section presents the methodology of text selection procedure for phonetically balanced speech corpus. We measured the phonetic coverage based on biphone in the phonetically transcribed database. In addition to phonetic coverage, we have also tried to maintain a prosodic and syntactic variation in the corpus for future works. Some sparse work has been done on phontactic constraint of Bangla. This is one of the obstacles to define an optimized biphone list. Therefore, we used the whole list of biphones in this study. Here, optimized means phonetically constrained biphones need to be omitted from the list. The system diagram of the whole development process is shown in figure 1.

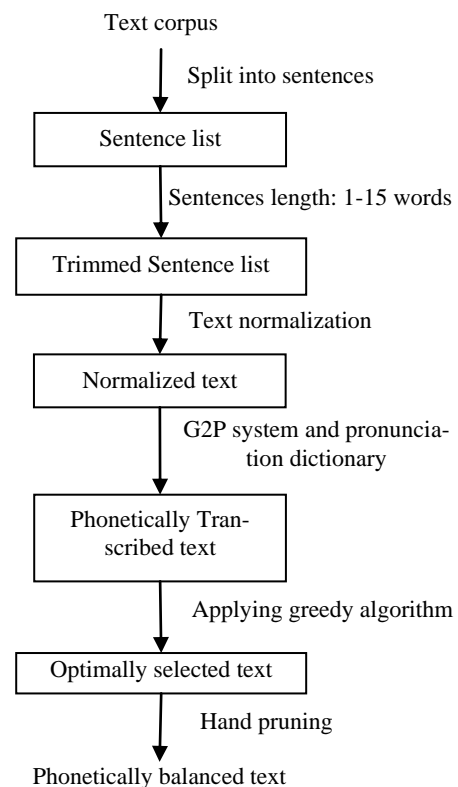


Figure 1: System diagram of phonetically balanced text selection

3.1 Text collection and normalization

Text selection from various domains is one of the most frequently used techniques for development of a speech corpus. However, it is one of the most time consuming phase, since a lot of manual work needs to be done, such as selecting different categories of text, proof reading and manual correction after text normalization. Therefore some constrains were considered dur-

ing text selection. The text was collected from two different sources such as prothom-alo news corpus (Yeasir et al., 2006) and CRBLP speech corpora (Firoj et al., 2010). The prothom-alo news corpus has 39 categories of text such as – general news, national news, international news, sports, special feature, editorial, sub-editorial and so on. Table 1 shows the frequency analysis of the initial text corpus.

Corpus	Sentences	Total to- kens	Token type
Prothom-alo news corpus	2,514,085	31,158,189	529,620
CRBLP read speech corpus	10,896	1,06,303	17,797
Total:	2,524,981	31,264,492	547,417

Table 1: Frequency distribution of the corpus

Starting with our initial text corpus consisting ~31 millions words and ~2.5 millions sentences we used a python script to split the corpus into sentences based on punctuation marks such as ?, | and !. It is observed that the length of some sentences is too long i.e. more than 15 words, even more than 25 words. Study of Kominek and Black, (2004) explained and our recording experience claimed that, sentences longer than 15 words are difficult to read aloud without making a mistake. With respect to the length, we maintained the constraints and filtered out sentences that are not between 1-20 words. Table 2 shows the frequency analysis of the corpus after the filtering.

Corpus	Sentences	Total to- kens	Token type
Prothom-alo news corpus	2,014,032	21,177,137	487,158
CRBLP read speech corpus	9,130	68,306	13,270
Total:	2,023,162	21,245,443	500,428

Table 2: Frequency distribution of the corpus after filtering

The text contains large number of non-standard words (NSW) (Sproat et al., 2008) such as number, date and phone number which need to be normalized to get full form of the NSW's. It is then normalized using a text normalization tool Firoj et al., (2009). There are some ambiguous NSW tokens such as year-number and time-floating point number. For example: (i). the token ১৯৯৯ (1999) could be considered as a year

and at the same time it could be considered as number and (ii). the token ১২.৮০ (12.80) could be considered as a floating point number or it could be considered as a time. In case of these ambiguous tokens the accuracy of the tool is 87% (Firoj et al., 2009). The 13% error was solved in the final text selection procedure. On the other hand, the accuracy rate of non-ambiguous token is 100% such as date (e.g: ০২-০৬-২০০৬), range (e.g: ১০-১২), ratio (e.g: ১/২), roman (e.g: I, II.), ordinal number (e.g: ১ম, ২য়, ৩য়) and so on.

Example of token	Normalized form
১২১	একশত একুশ
১ম	প্রথম
০২৯৫৬৭৪৪৭	শূন্য দুই নয় পাঁচ ছয় সাত চার চার সাত

3.2 Phoneme set

Defining the phoneme set is important for a phonetically balanced corpus. We considered the biphone as a unit for phonetically balanced corpus. The phoneme inventory used in this study is the one found in Firoj et al., (2008 a) and Firoj et al., (2008 b). The phoneme inventory consists of 30 consonants and 35 vowels including diphthongs. Since a biphone is the combination of two phones and Bangla phone inventory has 65 phones (Firoj et al., 2008 a) (Firoj et al., 2008 b), so the total number of biphones consist $65 \times 65 = 4225$. However, all biphones would not belong to the language in terms of phonotactic constraints.

Since no notable work has been done on phonetic constraint and as it is beyond our scope, we have not optimized the biphone list in this work.

3.3 Phonetic transcription

The phonetically transcribed text is needed to represent the phonetic coverage. Therefore, the text has to be phonetized so that the distribution of the phonetic unit can be analyzed. To perform phonetic transcription each sentence is tokenized based on 'white space' character. Then each word is passed through the CRBLP pronunciation lexicon (2009) and a G2P (Grapheme to Phoneme) converter (Ayesha et al., 2006). The system first checks the word in the lexicon, if the word is not available in the lexicon then it is passed through the G2P system for phonetic transcription. The CRBLP pronunciation lexicon contains 93K entries and the accuracy of the G2P system is 89%. So there could be errors in pho-

netic transcription due to the low accuracy rate of G2P system which is unavoidable. Manual correction of every word is not practical so a decision had been made that this problem would be solved in the "hand pruning" stage. In phonetic transcription we used IPA⁷, since IPA has been standardized as an internationally accepted transcription system. A phonetic measurement has been conducted after the text has been phonetically transcribed. The phonetic measurement of phone, biphone and triphone is shown in table 3.

Pattern type	Unique	Total in the corpus
Phones	65	119,068,607 (~119 millions)
Biphones	3,277	47,360,819 (~47 millions)
Triphones	274,625	115,048,711 (~115 millions)

Table 3: Phonetic measurement of the corpus

Though the corpus contains all the phones, it does not cover all the biphones. There could be several reasons:

1. Trimming the main sentence list to 20 words per sentence.
2. The frequency of these missing biphones is too low in the spoken space of this language.

3.4 Balanced Corpus Design

The greedy selection algorithm (Santen and Buchsbaum, 1997) has been used in many studies of the corpus design. This is an optimization technique for constructing a subset of sentences from a large set of sentences to cover the largest unit space with the smallest number of sentences. Prior to the selection process, the target space i.e. biphone is defined by the unit definition, mainly the feature space of a unit. A detail of the algorithm is shown below:

Algorithm

Step 1: Generate a unique biphone list from the corpus.

Step 2: Calculate frequency of the biphone in the list from the corpus.

Step 3: Calculate weight of each biphone in the list where weight of a biphone is inverse of the frequency.

Step 4: Calculate a score for every sentence. The sentence score is defined by the equation (1).

Step 5: Select the highest scored sentence.

Step 6: Delete the selected sentence from the corpus.

Step 7: Delete all the biphones found in the selected sentence from the biphone list.

Step 8: Repeat from Step 2 to 6 until the biphone list is empty.

$$Score = \left(\sum_i^{N_p} \left\{ \frac{1}{Pfi} \right\} \right)$$

Equation 1: Sentence score

SC - sentence score

N_p - the number of phonemes in each sentence

Pfi - the i^{th} phoneme frequency of the sentence in the corpus.

This algorithm successively selects sentences. The first sentence is the sentence, which cover largest biphone count.

Our text corpus contains ~2 millions sentences with 47 millions biphones. Based on experiment, it took 26 hours 44 mins of CPU time in a Core i5 due 2.4 GHz PC equipped with 3 GB memory to run the greedy selection process.

It is observed that, the results of automatic selection are not ideal due to accuracy rate of text normalizer and G2P system. So a hand pruning (Kominek and Black, 2004) is required. A visual inspection was made by considering several criteria such as awkward grammar, confusable homographs and hard to pronounce words. Next, the phonetically transcribed text was visually inspected, as our text normalization and G2P system produced some errors. Finally, a phonetically balanced text corpus was developed.

4 Recording

The next issue is the recording, which relates to selecting speaker, recording environment and recording instrument. Since speaker choice is perhaps one of the most vital areas for recording so a careful measure was taken. A female speaker was chosen who is a professional speaker and aged 29.

As far as recording conditions are concerned, we tried to maintain as high quality as possible. The recording of the utterances was done using the Nundo speech processing software. A professional voice recording studio was chosen to record the utterances. The equipment consisted of an integrated Tascam TM-D4000 Digital-Mixer, a high fidelity noise free Audiotechnica microphone and two high quality multimedia speaker systems. The voice talent was asked to keep a distance of 10-12 inches from the microphone. Optionally a pop filter was used between

⁷ IPA- International Phonetic alphabet

the speaker and the microphone to reduce the force of air puffs from bilabial plosive and other strongly released stops. The speech data was digitized at a sample rate 44.1 kHz, sample width 24-bit resolution and stored as wave format. After each recording, the moderator checked for any misleading pronunciation during the recording, and if so, the affected utterances were re-recorded.

There were a few challenges in the recording. First, speaker was asked to keep the speaking style consistent. Second, speaker was supervised to keep the same tone in the recording. Since speaking styles vary in different sessions, monitoring was required to maintain the consistency. To keep the consistency of the speaking style, in addition to Zhu et al. [29] recommendation the following specifications were maintained:

1. Recording were done in the same time slot in every session i.e 9.00 am to 1.00 pm.
2. A 5 minutes break was maintained after each 10 minutes recording.
3. Consistent volume of sound.
4. Normal intonation was maintained without any emotion.
5. Accurate pronunciation.

Pre-recorded voice with appropriate speaking style was used as a reference. In each session, speaker was asked to adjust his speaking style according to the reference voice.

5 Annotation

The un-cleaned recorded data was around 2 hours 5 minutes and it had a lot of repetition of the utterances. So in annotation, the recorded wave was cleaned manually using a proprietary software wavlab which tends to reduce the recorded data to 1 hours 11 minutes. Then, it was labeled (annotated) based on id using praat (Firoj et al., 2010). Praat provides a textgrid file which contains labels (in our case it is wave id) along with start and end time for each label. A separate praat script was written to split the whole wave into individual wave based on id with start and end time. We used id instead of text in labeling.

The structure of the corpus was constructed in a hierarchical organization using the XML standard. The file contains meta-data followed by data. The metadata contains recording protocol, speaker profile, text, annotation and spoken content. The data contains sentences with id, orthographic form, phonetic form and wave id.

6 Results

It was our desire to design a speech corpus which will exhibit good biphone coverage. The information of phonetic coverage of the corpus is shown in table 4.

Pattern type	No of unique units	Total found in the corpus (unique)	Coverage
Phones	65	65	100%
Biphones	4225	3,277	77.56%
Triphones	274,625	13911	5.06%

Table 4: Phonetic coverage information of the corpus

The percentages for biphone and triphone coverage are based on a simple combination. Thus the number of possible biphone is 4,225 and triphones is 274,625. This corpus covers nearly 100% phone and 77.56% biphone. A natural speech synthesizer achieved with 80% coverage of biphone as shown in Arctic of Kominek and Black, (2004). So it is hoped that better speech applications could be achievable by using this corpus.

We have also performed a frequency analysis of the phonemes on the whole phonetic corpus. And during the analysis, we observed an interesting phenomenon about the missing 22.44% biphone. That is, the phonemes whose frequency is less frequent (<0.11%) in the phonetic corpus, and surprisingly their combination came in the missing 22.44% biphone list. Observation also says that this combination basically came from diphthongs and a few nasals. So based on our empirical study we can claim that our findings of speech corpus are balanced and it would cover all of our everyday spoken space. This analysis leads to another assumption that this missing biphone list could be part of phonotactic constraint of Bangla language. This means that this combination possibly, may never occur in the spoken space of Bangla language. An effort needs to be done about the missing diphone using dictionary words and linguistic experts.

6.1 Comparison between festvox and our approach

We have made a comparison between the techniques that is used in festvox “nice utterance selection” (Kominek and Black, 2004)(Black and Lenzo, 2000) and the approach used in S. Kasuriya et al., (2003) and Patcharika et al., (2002) that we followed in this study. The difference

between these two approaches is that festvox technique uses most frequent words to select the text in the first step. The rest of the steps are the same in both approaches. We experimented festvox nice utterance selection tool (*text2utts*) using our data and got the result that is shown in table 5. A comparison between two techniques is shown in table 5. According to Firoj et al., (2008 a) and Firoj et al., (2008 b) Bangla has 65 phonemes including vowels and consonants. This leads to $65 \times 65 = 4225$ biphones in Bangla. So in selecting corpus, a corpus would be best when it covers maximum number of biphones. So in festvox approach, the selected corpus covers 1,495 biphones which is 35.38% of the whole biphone set. On the other hand in our approach we got 3277 biphones of the selected text which results 77.56% coverage.

Here, in festvox experiment we used most frequent 50,000 words in the first step to select the text. This approach limits the festvox *text2utts* tool to select the maximum number of utterances.

Pattern	Festvox approach	Approach used in this study
No. of sentences	677	977
No. of biphones	26,274	70,030
No. of phones	26,914	71,007
Biphone coverage	35.38% (1495 biphones)	77.56% (3276 biphones)
Phone coverage	84.61% (55 phones)	100% (65 phones)

Table 5: Comparison between festvox and our technique

7 Conclusion and future remarks

In this paper we presented the development of a phonetically balanced Bangla speech corpus. This speech corpus contains 977 sentences with 77.56% biphone coverage. It needs more sentences to cover all biphones. To do that, more text corpora may be required. However, finding out all biphones is pragmatically impossible due to the linguistic diversity and phonotactic constraint of a language. Besides, a significant amount of effort is needed to be able to use this resource in real speech applications. The efforts include recording voices by number of male and female voice talents for speech synthesis in a professional recording environment. Speech recognition application requires more recording data in different environments which includes record-

ing the voice by huge number (>50) of male and female voice talents.

References

- A. W. Black and K. Lenzo, 2000. *Building voices in the Festival speech synthesis system*, <http://festvox.org/bsv>.
- Ayesha Binte Mosaddeque, Naushad UzZaman and Mumit Khan, 2006. *Rule based Automated Pronunciation Generator*, Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh, December
- C. W. Patcharika, C. Wutiwitsachai, P. Cotsomrong, and S. Suebvisai, 2002. *Phonetically distributed continuous speech corpus for thai language*, Available online at: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.7778>
- CRBLP pronunciation lexicon, 2009. CRBLP, Available: <http://crblp.bracu.ac.bd/demo/PL/>
- Dafydd Gibbon, Inge Mertins, Roger K. Moore, 2000. *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation* (The Springer International Series in Engineering and Computer Science), Springer, August 31,
- Firoj Alam, S. M. Murtoza Habib and Professor Mumit Khan, 2008 a. *Acoustic Analysis of Bangla Consonants*, Proc. Spoken Language Technologies for Under-resourced language (SLTU'08), Vietnam, May 5-7, page 108-113.
- Firoj Alam, S. M. Murtoza Habib, and Mumit Khan, 2008 b. *Research Report on Acoustic Analysis of Bangla Vowel Inventory*, Center for Research on Bangla Language Processing, BRAC University,
- Firoj Alam, S. M. Murtoza Habib and Mumit Khan, 2009. *Text Normalization System for Bangla*, Conference on Language and Technology 2009 (CLT09), NUCES, Lahore, Pakistan, January 22-24,
- Firoj Alam, S. M. Murtoza Habib, Dil Afroza Sultana and Mumit Khan, 2010. *Development of Annotated Bangla Speech Corpora*, Spoken Language Technologies for Under-resourced language (SLTU'10), Universiti Sains Malaysia, Penang, Malasia, May 3 - 5,
- Fisher, William M.; Doddington, George R. and Goudie Marshall, Kathleen M. 1986. *The DARPA Speech Recognition Research Database: Specifications and Status*, Proceedings of DARPA Workshop on Speech Recognition. pp. 93-99.
- François, H. and Boëffard, O., 2002. *The Greedy Algorithm and its Application to the Construction of*

- a Continuous Speech Database* , Proc. of LREC, Las Palmas de Gran Canaria, Spain,
- François, H. and Boëffard, O., 2001. *Design of an Optimal Continuous Speech Database for Text-To-Speech Synthesis Considered as a Set Covering Problem* , Proc. of Eurospeech, Aalborg, Denmark,
- Gibbon, D., Moore, R., Winski, R. (eds.), 1997. *Handbook of Standards and Resources for Spoken Language Systems* , Mouton de Gruyter, Berlin New York
- J. Kominek and A. Black, 2004. *The cmu arctic speech databases* , 5th ISCA Speech Synthesis Workshop, pp. 223-224,
- S. Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, S. Kanokphara, and N. Thatphithakkul, 2003. *Thai speech corpus for thai speech recognition* , The Oriental COCODA 2003, pp. 54-61. [Online]. Available online at:
http://www.tcllab.org/virach/paper/virach/colips2004_final.rtf
- Sproat R., Black A., Chen S., Kumar S., Ostendorf M., and Richards C, 2008. *Normalization of Non-Standard Words: WS'99 Final Report* , CLSP Summer Workshop, Johns Hopkins University, 1999, Retrieved (June, 1, 2008). Available: www.clsp.jhu.edu/ws99/projects/normal
- V. Radová and P. Vopálka, 1999. *Methods of sentences selection for read-speech corpus design* , in TSD '99: Proceedings of the Second International Workshop on Text, Speech and Dialogue. London, UK: Springer-Verlag, pp. 165-170. [Online]. Available:
<http://portal.acm.org/citation.cfm?id=720594>
- Van Santen, J P. H. and Buchsbaum, A. L., 1997. *Methods for optimal text selection* , Proc. of Eurospeech, p. 553-556, Rhodes, Greece,
- Yeasir Arafat, Md. Zahurul Islam and Mumit Khan, 2006. *Analysis and Observations From a Bangla news corpus* , Proc. of 9th International Conference on Computer and Information Technology (ICIT 2006), Dhaka, Bangladesh, December
- Yoshida Yoshio, Fukuroya Takeo, Takezawa Toshioyuki, 2002. *ATR Speech Database* , Proceedings of the Annual Conference of JSAI, VOL.16th, 124-125, Japan

HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya

Michael Gasser

Indiana University

Bloomington, Indiana, USA

gasser@cs.indiana.edu

Abstract

Despite its linguistic complexity, the Horn of Africa region includes several major languages with more than 5 million speakers, some crossing the borders of multiple countries. All of these languages have official status in regions or nations and are crucial for development; yet computational resources for the languages remain limited or non-existent. Since these languages are complex morphologically, software for morphological analysis and generation is a necessary first step toward nearly all other applications. This paper describes a resource for morphological analysis and generation for three of the most important languages in the Horn of Africa, Amharic, Tigrinya, and Oromo.

1 Language in the Horn of Africa

The Horn of Africa consists politically of four modern nations, Ethiopia, Somalia, Eritrea, and Djibouti. As in most of sub-Saharan Africa, the linguistic picture in the region is complex. The great majority of people are speakers of Afro-Asiatic languages belonging to three sub-families: Semitic, Cushitic, and Omotic. Approximately 75% of the population of almost 100 million people are native speakers of four languages: the Cushitic languages Oromo and Somali and the Semitic languages Amharic and Tigrinya. Many others speak one or the other of these languages as second languages. All of these languages have official status at the national or regional level.

All of the languages of the region, especially the Semitic languages, are characterized by relatively complex morphology. For such languages, nearly all forms of language technology depend on the existence of software for analyzing and generating word forms. As with most other sub-Saharan languages, this software has previously

not been available. This paper describes a set of Python programs called HornMorpho that address this lack for three of the most important languages, Amharic, Tigrinya, and Oromo.

2 Morphological processing

2.1 Finite state morphology

Morphological analysis is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes. **Morphological generation** is the reverse process. Both processes relate a **surface** level to a **lexical** level. The relationship between the levels has traditionally been viewed within linguistics in terms of an ordered series of phonological rules.

Within computational morphology, a very significant advance came with the demonstration that phonological rules could be implemented as **finite state transducers** (Kaplan and Kay, 1994) (FSTs) and that the rule ordering could be dispensed with using FSTs that relate the surface and lexical levels directly (Koskenniemi, 1983), so-called “two-level” morphology. A second important advance was the recognition by Karttunen et al. (1992) that a cascade of composed FSTs could implement the two-level model. This made possible quite complex finite state systems, including ordered **alternation rules** representing context-sensitive variation in the phonological or orthographic shape of morphemes, the **morphotactics** characterizing the possible sequences of morphemes (in canonical form) for a given word class, and a **lexicon**. The key feature of such systems is that, even though the FSTs making up the cascade must be composed in a particular order, the result of composition is a single FST relating surface and lexical levels directly, as in two-level morphology. Because of the invertibility of FSTs, it is a simple matter to convert an analysis FST (surface input

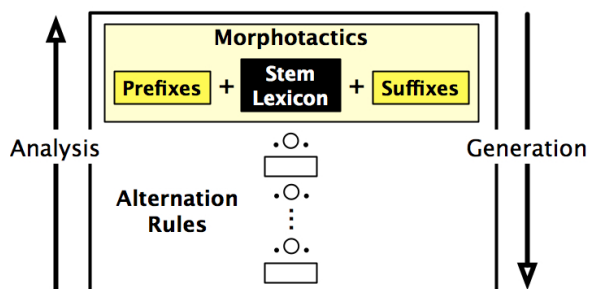


Figure 1: Basic architecture of lexical FSTs for morphological analysis and generation. Each rectangle represents an FST; the outermost rectangle is the full FST that is actually used for processing. “.o.” represents composition of FSTs, “+” concatenation of FSTs.

to lexical output) to one that performs generation (lexical input to surface output).

This basic architecture, illustrated in Figure 1, consisting of a cascade of composed FSTs representing (1) alternation rules and (2) morphotactics, including a lexicon of stems or roots, is the basis for the system described in this paper.

We may also want to handle words whose roots or stems are not found in the lexicon, especially when the available set of known roots or stems is limited. In such cases the lexical component is replaced by a phonotactic component characterizing the possible shapes of roots or stems. Such a “guesser” analyzer (Beesley and Karttunen, 2003) analyzes words with unfamiliar roots or stems by positing *possible* roots or stems.

2.2 Semitic morphology

These ideas have revolutionized computational morphology, making languages with complex word structure, such as Finnish and Turkish, far more amenable to analysis by traditional computational techniques. However, finite state morphology is inherently biased to view morphemes as sequences of characters or phones and words as concatenations of morphemes. This presents problems in the case of **non-concatenative morphology**, for example, discontinuous morphemes and the **template morphology** that characterizes Semitic languages such as Amharic and Tigrinya. The stem of a Semitic verb consists of a **root**, essentially a sequence of consonants, and a **template** that inserts other segments between the root consonants and possibly copies certain of the consonants. For example, the Amharic verb root *sbr*

‘break’ can combine with roughly 50 different templates to form stems in words such as ቤሱብረል *yi-sebr-al* ‘he breaks’, ተሰበረ *tesebber-e* ‘it was broken’, ላሱብረው *l-assebbir-ew*, ‘let me cause him to break something’, ሰበረ *sebabar-i* ‘broken into many pieces’.

A number of different additions to the basic FST framework have been proposed to deal with non-concatenative morphology, all remaining finite state in their complexity. A discussion of the advantages and drawbacks of these different proposals is beyond the scope of this paper. The approach used in our system is one first proposed by Amtrup (2003), based in turn on the well studied formalism of weighted FSTs. In brief, in Amtrup’s approach, each of the arcs in a transducer may be “weighted” with a feature structure, that is, a set of grammatical feature-value pairs. As the arcs in an FST are traversed, a set of feature-value pairs is accumulated by unifying the current set with whatever appears on the arcs along the path through the transducer. These feature-value pairs represent a kind of memory for the path that has been traversed but without the power of a stack. Any arc whose feature structure fails to unify with the current set of feature-value pairs cannot be traversed.

The result of traversing such an FST during morphological analysis is not only an output character sequence, representing the root of the word, but a set of feature-value pairs that represents the grammatical structure of the input word. In the generation direction, processing begins with a root and a set of feature-value pairs, representing the desired grammatical structure of the output word, and the output is the surface wordform corresponding to the input root and grammatical structure. In Gasser (2009) we showed how Amtrup’s technique can be applied to the analysis and generation of Tigrinya verbs. For an alternate approach to handling the morphotactics of a subset of Amharic verbs, within the context of the Xerox finite state tools (Beesley and Karttunen, 2003), see Amsalu and Demeke (2006).

Although Oromo, a Cushitic language, does not exhibit the root+template morphology that is typical of Semitic languages, it is also convenient to handle its morphology using the same technique because there are some long-distance dependencies and because it is useful to have the grammatical output that this approach yields for analysis.

3 HornMorpho

HornMorpho is a set of Python programs for analyzing and generating words in Amharic, Tigrinya, and Oromo. A user interacts with the programs through the Python interpreter. HornMorpho is available for download, under the GPL3 license, at <http://www.cs.indiana.edu/~gasser/Research/software.html>. Complete documentation is included with the downloaded archive.

For each language, HornMorpho has a lexicon of verb roots and (except for Tigrinya) noun stems.¹ For Amharic, the lexicon is derived from the Amharic-English dictionary of Aklilu (1987), which is available under the Creative Commons Attribution-Noncommercial 3.0 United States License at <http://nlp.amharic.org/resources/lexical/word-lists/dictionaries/>; there are currently 1,851 verb roots and 6,471 noun stems. For Oromo the lexicon of verb and noun roots is extracted from the dictionaries of Gragg (1982) and Bitima (2000); there are currently 4,112 verb roots and 10,659 noun stems. For Tigrinya, the lexicon of verb roots is derived from Efreim Zacarias' (2009) online dictionary, accessible at <http://www.memhr.org/dic/>; there are currently 602 verb roots.

3.1 System architecture

The full morphology processing system (see Figure 1) consists of analysis and generation FSTs for each language. For Amharic and Tigrinya there are separate lexical and “guesser” FSTs for each processing direction. Verbs (all three languages) and nouns (Amharic and Oromo only) are handled by separate FSTs. Amharic and Oromo have separate FSTs for verb segmentation, as opposed to grammatical analysis, and Amharic has a separate FST for orthography-to-phonology conversion.

Each of these FSTs in turn results from the composition of a cascade of simpler FSTs, each responsible for some aspect of morphology. The most complicated cases are Amharic and Tigrinya verbs, which we discuss in more detail in what follows and illustrate in Figure 2. At the most abstract (lexical) end is the heart of the system, the morphotactic FST. Most of the complexity is

¹ Amharic adjectives, which behave much like nouns, are grouped with nouns in the system.

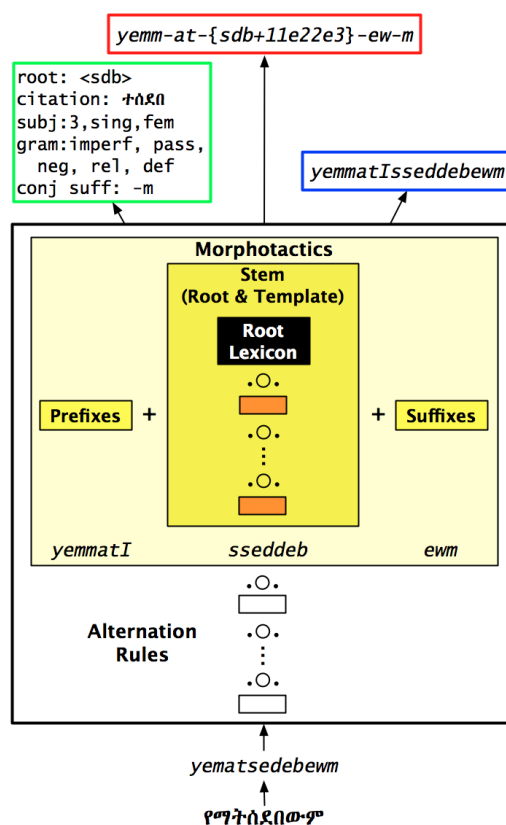


Figure 2: Architecture of Amharic verb analysis FST. Shown: analysis of the verb የማትሰደበውም ‘who (she) is also not insulted’. Output analyses: anal_word (green border); seg_word (red border); phon_word (blue border).

within the FST responsible for the stem. The stem FST is composed from a sequence of five simpler FSTs representing stem-specific alternation rules and the legal sequences of consonants and vowels making up the stem. For the lexical, but not the guesser, FSTs, a further FST containing the lexicon of known roots is part of the stem cascade.

Prefix and suffix FSTs are concatenated onto the stem FST to create the full verb morphotactic FST. The remaining FSTs (15 of them for Amharic verbs, 17 for Tigrinya verbs) implement alternation rules that apply to the word as a whole, including allomorphic rules and general phonological or orthographic rules.

The figure shows the analysis of the Amharic verb የማትሰደበውም yemmatisseddəbəwm ‘who (she) is also not insulted’. The word is input in the Ge’ez orthography used for Amharic and Tigrinya. This writing system fails to indicate consonant gemination as well as the epenthetic vowel *i*, which is introduced to break up some conso-

nant clusters in both languages. Gemination is extremely important for natural speech synthesis in Amharic and Tigrinya, so it is crucial to be able to restore it in text-to-speech applications. There is in fact relatively little ambiguity with respect to gemination, but gemination is so tied up with the morphology that a relatively complete morphological analyzer is necessary to perform the restoration. HornMorpho has this capacity.

The word is first romanized to *yematsedebewm*.² At this stage, none of the consonants is geminated, and the epenthetic vowel is missing in the romanized form. Processing is then handled by the single analysis FST, but to understand what goes on, it is still to convenient to think of the process in terms of the equivalent cascade of simpler FSTs operating in sequence. The first FST in the cascade performs orthographic-to-phonological conversion, resulting in all possible pronunciations of the input string, including the correct one with the appropriate consonants geminated. This form and the other surviving strings are processed by the intervening phonological FSTs, each responsible for an alternation rule. Among the strings that survive as far as the morphotactic FST is the correct string, *yemmatIssedebewm*, which is analyzable as the pair of prefixes *yemm-at*, the stem *sseddeb*, and the pair of suffixes *ew-m*.

The stem is processed by the stem FST, which extracts the root, *sdb*, and various grammatical properties, including the fact that this form is passive. The lexical analyzer includes all of the verb roots known to the system within the stem FST, whereas the guesser analyzer includes only information about sequences of consonants making up possible roots in the language. For example, if the consonant sequence *s,d,b* in the original word were replaced by a fictitious root *mbz*, the guesser analyzer (but not the lexical analyzer) would posit this as the root of the word. The final output analyses are shown at the top of Figure 2. The three possibilities correspond to three different HornMorpho functions, discussed in the next section.

3.2 Functions

Each of the functions for morphological analysis has two versions, one for analyzing single words, the other for analyzing all of the words in a

²HornMorpho uses an ASCII romanization scheme developed by Firdyiwek and Yaqob (1997).

file. The functions `anal_word` and `anal_file` take input words and output a root or stem and a grammatical analysis. In Figure 2, the output of `anal_word` is outlined in green. The input word has the root *sdb* ‘insult’ and the citation form ተሰደበ; has a third person singular feminine subject; is in the imperfective tense/aspect; is relativized, negative, and definite; and has the conjunctive suffix *-m*.

For Amharic and Oromo, there are two additional analysis functions, `seg_word` and `seg_file`, which segment input verbs (also nouns for Oromo) into sequences of morphemes. In the example in Figure 2, the output of `seg_word` is shown in red. The constituent morphemes are separate by hyphens, and the stem is enclosed in brackets. The root and template for the stem are separated by a plus sign. The template notation *11e22e3* indicates that the first and second root consonants are geminated and the vowel *e* is inserted between the first and second and second and third root consonants.

For Amharic only, there are further functions, `phon_word` and `phon_file`, which convert the input orthographic form to a phonetic form as would be required for text-to-speech applications. In the figure, the output of `phon_word` is outlined in blue. Three of the consonants are geminated, and the epenthetic vowel (romanized as *I*) has been inserted to break up the cluster *tss*.

Below are more examples of the analysis functions, as one would call them from the Python interpreter. Note that all of the HornMorpho functions take a first argument that indicates the language. Note also that when a wordform is ambiguous (Example 3), the analysis functions return all possible analyses.

Example 1 `anal_word` (Tigrinya)

```
>>> anal_word('ti', 'ተሰደበግግግ')
Word: ተሰደበግግግ
POS:verb, root:<gTm>, cit:አጋጠጠ
subject: 3, sing, masc
object: 1, plur
grammar: imperf, recip, trans, rel
preposition: bI
```

Example 2 `seg_word` (Oromo)

```
>>> seg_word('om', 'dhukkubdi')
dhukkubdi: dhukkub-t-i
```

There is a single function for generation, `gen`, which takes a stem or root and a set of grammat-

Example 3 `phon_word` (Amharic)

```
>>> phon_word('am', '፳፻፳፻')
yImetallu yImmetallu
```

ical features. For each part of speech, there is a default set of features, and the features provided in the function call modify these. In order to use `gen`, the user needs to be familiar with the HornMorpho conventions for specifying grammatical features; these are described in the program documentation.

With no grammatical features specified, `gen` returns the canonical form of the root or stem, as in the Oromo example 4 (*sirbe* is the third person singular masculine past form of the verb). Example 5 is another Oromo example, with additional features specified: the subject is feminine, and the tense/mood is present rather than past.

Example 4 `gen` (Oromo 1)

```
>>> gen('om', 'sirb')
sirbe
```

Example 5 `gen` (Oromo 2)

```
>>> gen('om', 'sirb', '[sb=[+fem],tm=prs]')
sirbiti
```

4 Evaluation

Evaluating HornMorpho is painstaking because someone familiar with the languages must carefully check the program's output. A useful resource for evaluating the Amharic and Tigrinya analyzers is the word lists compiled by Biniam Gebremichael's web crawler, available on the Internet at <http://www.cs.ru.nl/~biniam/geez/crawl.php>. The crawler extracted 227,984 unique Tigrinya wordforms and 397,352 unique Amharic wordforms.

To evaluate the Amharic and Tigrinya analyzers in HornMorpho, words were selected randomly from each word list, until 200 Tigrinya verbs, 200 Amharic verbs, and 200 Amharic nouns and adjectives had been chosen. The `anal_word` function was run on these words, and the results were evaluated by a human reader familiar with the languages. An output was considered correct only if it found all legal combinations of roots and grammatical structure for a given wordform and included no incorrect roots or structures. The program made 8 errors on the Tigrinya verbs (96%

accuracy), 2 errors on the Amharic verbs (99% accuracy), and 9 errors on the Amharic nouns and adjectives (95.5% accuracy).

To test the morphological generator, the `gen` function was run on known roots belonging to all of the major verb root classes.³ For each of these classes, the program was asked to generate 10 to 25 verbs depending on the range of forms possible in the class, with randomly selected values for all of the different dimensions, a total of 330 tests. For Amharic, the program succeeded on 100% of the tests; for Tigrinya it succeeded on 93%.

In all cases, the errors were the result of missing roots in the lexicon or bugs in the implementation of specific phonological rules. These deficiencies have been fixed in the most recent version of the program.

Although more testing is called for, this evaluation suggests excellent coverage of Amharic and Tigrinya verbs for which the roots are known. Verbs are the source of most of the morphological complexity in these languages. Nouns and adjectives, the only other words calling for morphological analysis, are considerably simpler. Because the plural of Tigrinya nouns is usually not predictable, and we have access to only limited lexical resources for the language, we have not yet incorporated noun analysis and generation for that language. For Amharic, however, the system is apparently able to at least analyze the great majority of nouns and adjectives. We treat all Amharic words other than verbs, nouns, and adjectives as unanalyzed lexemes.

For Oromo, the newest language handled by HornMorpho, we have not yet conducted a comparable evaluation. Any evaluation of Oromo is complicated by the great variation in the use of double consonants and vowels by Oromo writers. We have two alternatives for evaluation: either we make the analyzer more lenient so that it accepts both single and double vowels and consonants in particular contexts or we restrict the evaluation to texts that have been verified to conform to particular orthographic standards.

5 Conclusions and ongoing work

For languages with complex morphology, such as Amharic, Tigrinya, and Oromo, almost all computational work depends on the existence of tools for morphological processing. HornMorpho is a

³The Amharic noun generator has not yet been evaluated.

first step in this direction. The goal is software that serves the needs of developers, and it is expected that the system will evolve as it is used for different purposes. Indeed, some features of the Amharic component of the system have been added in response to requests from users.

One weakness of the present system results from the limited number of available roots and stems, especially in the case of Tigrinya. When a root is not known, the Tigrinya verb guesser analyzer produces as many as 15 different analyses, when in many cases only one of these contains a root that actually exists in the language. However, the guesser analyzer itself is a useful tool for extending the lexicon; when an unfamiliar root is found in multiple wordforms and in multiple morphological environments, it can be safely added to the root lexicon. We have explored this idea elsewhere (Gasser, 2010).

A more significant weakness of the analyzers for all three languages is the handling of ambiguity. Even when a root or stem is known, there are often multiple analyses, and the program provides no information about which analyses are more likely than others. We are currently working on extending the weighted FST framework to accommodate probabilities as well as feature structures on transitions so that analyses can be ranked for their likelihood.

Although Amharic and Tigrinya have very similar verb morphology, they are handled by completely separate FSTs in the current implementation. In future work we will be addressing the question of how to share components of the system across related languages and how to build on existing resources to extend the system to handle related Semitic (e.g., Tigre, Silt'e) and Cushitic (e.g., Somali, Sidama) languages of the region.

Finally, HornMorpho is designed with developers in mind, people who are likely to be comfortable interacting with the program through the Python interpreter. However, morphological analysis and generation could also be of interest to the general public, including those who are learning the languages as second languages. We are currently experimenting with more user-friendly interfaces. As an initial step, we have created a web application for analyzing and generating Tigrinya verbs, which is available here: <http://www.cs.indiana.edu/cgi-pub/gasser/L3/morpho/Ti/v/anal/>.

References

- Aklilu, A. (1987). *Amharic-English Dictionary*. Kuraz Printing Press, Addis Ababa.
- Amsalu, S. and Demeke, G. A. (2006). Non-concatenative finite-state morphotactics of Amharic simple verbs. *ELRC Working Papers*, 2(3).
- Amtrup, J. (2003). Morphology in machine translation systems: Efficient integration of finite state transducers and feature structure descriptions. *Machine Translation*, 18:213–235.
- Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications, Stanford, CA, USA.
- Bitima, T. (2000). *A dictionary of Oromo technical terms*. Rüdiger Köpper Verlag, Köln.
- Firdyiwek, Y. and Yaqob, D. (1997). The system for Ethiopic representation in ASCII. URL: citeseer.ist.psu.edu/56365.html.
- Gasser, M. (2009). Semitic morphological analysis and generation using finite state transducers with feature structures. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 309–317, Athens, Greece.
- Gasser, M. (2010). Expanding the lexicon for a resource-poor language using a morphological analyzer and a web crawler. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.
- Gragg, G. (1982). *Oromo dictionary*. Michigan State University Press, East Lansing, MI, USA.
- Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, 20:331–378.
- Karttunen, L., Kaplan, R. M., and Zaenen, A. (1992). Two-level morphology with composition. In *Proceedings of the International Conference on Computational Linguistics*, volume 14, pages 141–148.
- Koskenniemi, K. (1983). Two-level morphology: a general computational model for word-form recognition and production. Technical Report Publication No. 11, Department of General Linguistics, University of Helsinki.
- Zacarias, E. (2009). Memhr.org dictionaries (English-Tigrinya, Hebrew-Tigrinya dictionaries). Available at <http://www.memhr.org/dic/>.

Swahili Inflectional Morphology for the Grammatical Framework

Wanjiku Ng'ang'a

School of Computing & Informatics, University of Nairobi, Kenya

wanjiku.nganga@uonbi.ac.ke

Abstract

Grammatical Framework is a grammar formalism based on type theory and implemented in Haskell, that utilizes the interlingua approach to multilingual translation. Multilingualism is achieved by defining resource grammar libraries for each individual language within the framework. Here, we present the definition of the inflectional morphology grammars for Swahili, as part of the Swahili resource grammar library within the Grammatical Framework. In this framework, morphological constructs are implemented using the functional morphology approach which requires definition of linguistic data types and morphological functions.

1 Introduction

Grammatical Framework (GF) (Ranta, 2004) is a grammar formalism based on type theory. It has been designed to handle several languages in parallel. Its main feature is the separation of abstract and concrete syntax, which makes it very suitable for writing multilingual grammars. The abstract part of a grammar defines a set of abstract syntactic structures, called abstract terms or trees, while the concrete part defines a relation between abstract structures and concrete structures. Multilingual grammars formalize the notion that different languages share the same grammatical categories (e.g. noun phrases, verb phrases etc) and syntax rules (e.g. nominalization and predication). An abstract syntax in GF deals with language-independent (pure) tree structures while the language-dependent concrete syntax specifies how these trees are mapped into different languages. A multilingual GF grammar is therefore realised as a combination of an abstract syntax which can be mapped to a number of language-dependent concrete syntaxes, thereby achieving multilingualism.

To achieve multilingualism, GF uses a Resource Grammar (RG) library (Ranta, 2009), which is essentially a set of parallel grammars for differ-

ent languages. The grammar defines, for each language, a complete set of morphological paradigms and a syntax fragment. Currently, the library covers sixteen languages: Bulgarian, Catalan, Danish, Dutch, English, Finnish, French, German, Italian, Norwegian (bokmål), Polish, Romanian, Russian, Spanish, Swedish and Urdu. Grammars for other languages, including Swahili, are under development. The main purpose of this grammar library is to allow application developers (non-linguist programmers) to write domain-specific grammars to support varied, linguistically correct multilingual applications. This is in line with modern software engineering practice where software libraries, comprising of a number of specially written routines are used as 'helper' code that provide services to some other independent programs. GF's multilingual grammar library provides a reusable interface to the thousands of details involved in morphology, lexicon, inflection and syntax, facilitating the easy development of multilingual natural language engineering applications such as machine translation, multilingual generation, natural language interfaces, spoken dialogue systems etc. To date, GF has been used to build a wide range of applications such as an authoring and documentation system for the KeY software specification system (Burke and Johannisson, 2005), WebALT's mathematical exercise translator (Caprotti, 2006), and the TALK project on spoken dialogue systems (Ljunglöf et al., 2008), to list a few.

A GF Resource Grammar comprises of two parts: morphological paradigms and a syntax fragment. This paper describes the development of Swahili morphological paradigms as a first step in creating a Swahili GF Resource Grammar. This work represents the first attempt to extend GF with a Bantu language. Section 2 gives a brief summary of the Swahili language with emphasis on morphology, while the definition of the language-dependent modules for Swahili morphology in GF is covered in Section 3.

2 Swahili

Swahili is a Bantu language belonging to the Niger-Congo family. It is a highly inflecting language where both prefixed and suffixed morphemes play an important grammatical role. The functions of prefixes are particularly important in both nominal and verbal morphology. In the case of nouns, as is typical with Bantu languages, each noun belongs to a noun class which is signaled by a pair of prefixes attached to the nominal stem, denoting singular and plural forms. In addition, some nouns take an optional locative suffix e.g. *nyumba-ni* which means 'in the house', is obtained by adding the *-ni* locational suffix to the root *nyumba*. Verbs have an agglutinating structure where a system of affixes is used to mark various grammatical relations, such as subject, object, tense, aspect, and mood. There is a system of concordial agreement in which nouns must agree with the main verb of the sentence in class (and animacy) and number. Adjectives, possessive pronouns and demonstratives also agree in class and number with the noun they modify.

Swahili has a fairly fixed word order (SVO) at the sentence level, where the subject precedes the verb and the object, while within constituent phrases, modifiers succeed the head. Therefore adjectives, pronouns, determiners etc., follow the nouns they modify while adverbs come after the verb. Concrete grammars defined for Swahili should therefore capture these rich morphological and grammatical features of the language, to ensure generation of grammatically correct Swahili sentences.

3 Inflectional Morphology Paradigms for Swahili

Hurskainen (1992) has defined Swahili morphology using the two-level formalism, within the Swahili Language Manager (SALAMA) computational suite (Hurskainen, 1999). In contrast to finite-state approaches to morphology, GF uses a functional morphology approach to defining inflectional morphological paradigms. Rather than working with untyped regular expressions which is the state of the art of morphology in computational linguistics, functional morphology defines finite functions over hereditarily finite algebraic data types (Forsberg and Ranta, 2005). The definitions of these data types and functions form the language-dependent part of the morphology, while the language independent part consists of

an untyped dictionary format which is used for synthesis of word forms. These language-dependent data types and functions are organized into four grammar files: Res, Morpho, Cat and Paradigms. Construction of lexical entries relies on the language-dependent definitions, and the language-independent Lexicon file. In this paper, we describe the definition of Res, Morpho, Cat, Paradigms and Lexicon, for Swahili.

3.1 ResSwa

Res is a language-specific resource grammar that defines parameter types specific to a given language. These types denote linguistic features that cut across the language. For nouns, the following parameters are defined:

$$\text{Gender} = g1_2 \mid g3_4 \mid g5_6 \mid g5a_6 \mid g6 \mid g7_8 \mid g9_10 \mid g11 \mid g11_6 \mid g11_10 ;$$
$$\text{Case} = \text{Nom} \mid \text{Loc} ;$$
$$\text{Animacy} = \text{AN} \mid \text{IN} ;$$

The parameter Gender defines Swahili's noun classes where *g1_2* refers to noun class *m_wa* where the singular form begins with prefix *m-* and the plural with *wa-* e.g. *m-sichana* (girl) and *wa-sichana*. (girls). The parameter Case defines two options – nominative and locative to handle cases where some nouns optionally take a locational suffix as explained in section 2. The parameter Animacy, with values *animate* and *inanimate*, is defined to ensure correct subject-verb agreement for animate nouns that do not fall in the typical noun class *m-wa* for animates, but whose agreement features must match those of animate nouns. Examples of such nouns that do not fall in the typical animate class *m-wa*, include names of most animals as well as family relations e.g. *mama* 'mother', *baba* 'father', *dada* 'sister', and *ndugu* 'brother'. Another important parameter is Number (singular and plural) which is common to other languages and is therefore pre-defined elsewhere in the resource grammar library. In this work, we adopt the Swahili noun classes as motivated by Moxley (1998).

For verbs, the following parameters are defined:

$$\begin{aligned} \text{VForm} = & \\ & \text{Vinf} \mid \\ & \text{Vimper Number Person} \mid \\ & \text{VPres Number Gender Animacy Person} \mid \\ & \text{VPast Number Gender Animacy Person} \mid \\ & \text{VFut Number Gender Animacy Person}; \end{aligned}$$

The parameter Vform defines five different forms that a verb can take. The first form is the Infinitive form. Next, is the Imperative form which is dependent on Number and Person since this form is only applicable to second person singular (2SG) or plural (2PL) e.g. for the verb *cheza* 'to play', the imperative form for 2SG is *cheza* while that for 2PL is *chezeni*. The other three forms define Present, Past and Future tense, which are all dependent on Number, Gender, Animacy and Person, as shown by the examples in Table 1.

Gen-der	An-animacy	Num-ber	Per-son	Pre-fix (Present)	Pre-fix (Past)	Pre-fix (Future)
1_2	AN	SG	3	ana	ali	ata
1_2	AN	PL	3	wana	wali	wata
9_10	AN	SG	3	ana	ali	ata
9_10	IN	SG	3	ina	ili	ita

Table 1: Verbs

The parameter AForm has been defined for Adjectives:

AForm = AF Number Gender Animacy | AA ;

AForm is defined using two operations, AF and AA. AF is applicable for adjectives that agree with the number, gender and animacy of the modified noun, while AA is a defunct operation that can be used to handle exceptions. The Spatial parameter has been defined to distinguish between the proximal, distal and referential demonstratives e.g. *hii* 'this', *hiyo* 'that' and *ile* 'that' (distal), and has been defined as follows:

Spatial = SpHrObj | SpHr | HrObj ;

3.2 MorphoSwa

Morpho contains language-specific resource grammar constructs that define exactly how various inflectional morphological paradigms are realized. Morpho uses the parameters defined in Res to realize the categories defined in Cat, as detailed in section 3.3. The definitions contained in Morpho are typically functional (constructionist) which define how to inflect stems to obtain different word forms e.g Nouns, Verbs, Adjectives, demonstratives and pronouns, and are accessed via function calls from the Paradigms

grammar. Following is a GF code fragment that defines how to construct adjectives.

```
MkAdjective : Str -> Adj = \zuri -> {
  s = table {
    AF n g anim => case Predef.take 1 zuri of {
      "a|e|i|o|u" => VowelAdjprefix n g
        anim + zuri;
      _ => ConsonantAdjprefix n g anim + zuri
    };
    AA => zuri
  }
};
```

The above code defines a function mkAdjective which takes an adjective root e.g. *zuri* 'good' or *eupe* 'white' which is of type string (Str) and creates an Adjective of type Adj. If the adjectival root begins with a vowel, a semi-vowel or a consonant must be inserted at the end of the prefix. To enforce this requirement, the code defines two helper functions VowelAdjprefix and ConsonantAdjprefix to handle the two cases separately. Since the adjective prefix is dependent on the AForm parameter as described in section 3.1, each of the helper functions take the number (n), gender (g) and animacy (anim) of the modified noun as input to determine the adjective prefix and prepends the prefix to the adjectival root, thus forming the adjective. Table 1 shows how mkAdjective creates adjectives that adhere to Swahili morphological rules as described.

Gen-der	Anim-acy	Num-ber	Vowel root = zuri	Vowel root = eupe
1_2	Animate	SG	m-zuri	M-eupe => mw-eupe
9_10	Inanimate	SG	n-zuri	n-eupe => ny-eupe

Table 2: Adjectives

The following code fragment shows the definition of mkPronoun that takes Number and Person as input and generates a Pronoun:

```
mkPronoun : Number -> Person -> Str = \n,p ->
  case <n,p> of {
    <Sg,P1> => "mimi" ;
    <Sg,P2> => "wewe" ;
    <Sg,P3> => "yeye" ;
    <Pl,P1> => "sisi" ;
    <Pl,P2> => "nyinyi" ;
    <Pl,P3> => "wao" };
```

3.3 CatSwa

The Cat grammar defines all the lexical categories (closed, open and phrasal) that occur in language, and most of these are common to all languages. The Swahili concrete syntax file, CatSwa, currently defines the type specifications for the common base categories as they present in Swahili, as shown in Table 3.

Cat	Type	Example
N	Common Noun	<i>Msichana</i> 'Girl'
N2	Relational Noun	<i>Ndugu ya ..</i> 'Brother of ..'
Pron	Personal Pronoun	<i>Mimi</i> 'I'
V	One-place Verb	<i>Nitaimba</i> 'I will sing'
A	One-place Adjective	<i>Mzuri</i> 'Good'
Quant	Quantifier (Nucleus of Determiner)	<i>Hii/Hizi</i> 'This/These'
Prep	Preposition	<i>Ya</i> 'Of'
Num	Number	<i>Nne</i> 'Four'

Table 3: Lexical Categories

3.4 ParadigmsSwa

Paradigms defines the top level morphological functions that are used to construct lexical entries in the lexicon file, *LexiconSwa*. The functions correspond to the base lexical categories defined in CatSwa as shown in Table 3. Swahili Nouns are regular in the sense that given a noun root, gender, number and animacy, it is possible to use rules to generate a correct noun form. Hence, the abstract function *regN* is used to define common nouns of type N. In this definition, *Str* refers to the noun root which is passed to the *regN* function together with the values for Gender and Animacy:

regN : Str -> Gender -> Animacy -> N ;

The concrete form of *regN* calls the helper function *mkNomReg* that is defined in *MorphoSwa* to generate the final noun form. *mkNomReg* abstracts over the pre-defined Number parameter to generate both singular and plural forms of the noun root, and hence Number need not be passed directly to *regN*. Nouns of type N2 which take a

Noun of type N followed by a preposition, are constructed by the function *mkN2*:

mkN2 : N -> Prep -> N2 ;

Pronouns are defined by the function *mkPron*:

mkPron : Number -> Person -> Pron ;

The category Verb (V) is a simple form and is defined by the function *regV* which makes reference to the parameter *VForm* as described in section 3.1, to generate the infinitive, imperative, present, past and future tense forms for any regular verb. :

regV : Str -> V ;

Adjectives are constructed by the abstract function *regA* which also uses the type *AForm* to generate adjectives that conform to concordial agreement with the noun they modify:

regA : Str -> A ;

Prepositions are constructed by the abstract function *mkPrep* whose definitions is:

mkPrep : Str -> Prep ;

Quantifiers are constructed via the helper function *mkQuant* defined in *MorphoSwa*.

mkQuant : Spatial -> Number -> Gender ->
Animacy -> Case -> Person -> Str;

mkQuant takes as input the Spatial parameter that specifies whether to construct a proximal, distal or referential demonstrative. In addition, the number, gender and animacy values have to be specified since Swahili quantifiers must agree in number and gender with the modified noun, as shown by the examples in Table 4.

Gen-der	Anim-acy	Num-ber	Prox-imal	Distal	Refer-ential
1_2	AN	SG	Huyu	Huyo	Yule
1_2	AN	PL	Hawa	Hao	Wale
3_4	IN	SG	Huu	Huo	Ule
3_4	IN	PL	Hii	Hiyo	Ile
7_8	IN	PL	Hivi	Hivyo	Vile
7_8	AN	PL	Hawa	Hao	Wale

Table 4: Swahili Quantifiers

3.5 LexiconSwa

Lexicon is part of the top-level grammar in GF. This grammar defines and gives access to content words that can then be used by the syntax component of the resource grammar library. LexiconSwa uses the functions defined in ParadigmsSwa to define lexical entries (words) that conform to Swahili morphology. The content words defined here must be of the types defined in CatSwa. Currently, LexiconSwa contains 83 content words out of a total 300 words defined in the corresponding language-independent lexicon (abstract) file, Lexicon. Table 5 shows example lexical definitions from LexiconSwa, while Table 6 shows the corresponding definitions in the English lexicon, LexiconEng. The abstract function e.g. *country_N* is defined in the abstract grammar Lexicon while LexiconSwa defines the corresponding concrete syntax. For example, *country_N* is the abstract function defined in Lexicon, while the definition: *country_N = regN "nchi" e_e inanimate*; is its corresponding concrete linearization in Swahili. This definition states that to form the Swahili form for the noun 'Country', the function *regN* defined in ParadigmsSwa is called by LexiconSwa with the Swahili string for country, *nchi*, followed by the gender (*e_e* in this case) and animacy value (*inanimate*). The function *regN* then inflects the root appropriately to construct the singular and plural forms for *country_N* in Swahili.

Cat	Definition in LexiconSwa (Swahili)
N	<i>country_N = regN "nchi" e_e inanimate ;</i>
N	<i>cousin_N = regN "binamu" e_ma animate;</i>
N	<i>man_N = regN "mwanaume" m_wa animate ;</i>
N	<i>tree_N = regN "mti" m_mi inanimate ;</i>
N	<i>water_N = regN "maji" ma_ma inanimate ;</i>
V	<i>swim_V = regV "ogelea";</i>
A	<i>dirty_A = regA "chafu" ;</i>
N2	<i>father_N2 = mkN2 (regN "baba" e_e animate) (mkPrep "ya") ;</i>
Quant	<i>this_Quant = {s = \\n,g,anim,c => mkQuant SpHrObj n g anim Nom P3} ;</i>

Table 5: Swahili Lexicon Entries

Cat	Definition in LexiconEng (English)
N	<i>country_N = regN "country" ;</i>
N	<i>cousin_N = mkN human (regN "cousin") ;</i>
N	<i>man_N = mkN masculine (mk2N "man" "men") ;</i>
N	<i>tree_N = regN "tree" ;</i>
N	<i>water_N = regN "water" ;</i>
V	<i>swim_V = IrregEng.swim_V ;</i>
A	<i>dirty_A = regADeg "dirty" ;</i>
N2	<i>father_N2 = mkN2 (mkN masculine (mkN "father")) (mkPrep "of") ;</i>
Quant	<i>this_Quant = mkQuant "this" "these" ;</i>

Table 6: English Lexicon Entries

LexiconSwa and LexiconEng clearly demonstrate how GF achieves multilingualism by allowing languages to share an abstract syntax, but define language-dependent features in the concrete grammars. Table 7 shows example translations at the lexical level that have been generated automatically within GF.

Swahili	Abstract Function	English
<i>hawa</i>	<i>this_Quant</i>	<i>these</i>
<i>ndugu</i>	<i>brother_N2</i>	<i>brother/brothers</i>
<i>kiti</i>	<i>chair_N</i>	<i>chair</i>
<i>rafiki</i>	<i>rafiki_N</i>	<i>friend</i>
<i>marafiki</i>	<i>rafiki_N</i>	<i>friends</i>
<i>mwanaume</i>	<i>man_N</i>	<i>man</i>
<i>wanaume</i>	<i>man_N</i>	<i>men</i>
<i>mti</i>	<i>tree_N</i>	<i>tree</i>
<i>miti</i>	<i>tree_N</i>	<i>trees</i>
<i>vinalala</i>	<i>sleep_V</i>	<i>sleeping</i>
<i>wanawaza</i>	<i>think_V</i>	<i>thinking</i>
<i>nilitembea</i>	<i>walk_V</i>	<i>walked</i>
<i>watatembea</i>	<i>walk_V</i>	<i>walk</i>
<i>tulitembea</i>	<i>walk_V</i>	<i>walked</i>
<i>mrembo</i>	<i>beautiful_A</i>	<i>beautiful</i>
<i>warembo</i>	<i>beautiful_A</i>	<i>beautiful</i>
<i>kirembo</i>	<i>beautiful_A</i>	<i>beautiful</i>
<i>virembo</i>	<i>beautiful_A</i>	<i>beautiful</i>

Table 7: Parsing and Generation Examples

4 Conclusion

In this paper, we have described the first attempt to extend the morphological component of the Grammatical Framework with a Bantu language, Swahili. We have described the data types and corresponding functions that implement Swahili inflectional morphology following the functional morphology methodology. For the 83 content words that have been defined in the lexicon, it is possible to generate translations to the other 16 languages currently implemented in GF, and vice-versa. Subsequent development work will focus on defining all possible lexical categories in CatSwa, following the abstract grammar, Cat. We will then define more inflectional morphology functions to support the new category additions. Once the morphology paradigms are completed, we will define the core syntax grammars for Swahili, thereby facilitating the translation of full sentences and utterances from Swahili into all the other GF languages. This work will contribute greatly to the development of domain-specific HLTD applications that require localization and customization for a Swahili-speaking audience as posited by Ng'ang'a (2006). We also envisage generalizing the Swahili grammars to cater for a wide range of Bantu languages, by adopting the language family modular structure within GF that allows Romance languages to share a whole lot of data types and functions, and define only language-dependent exceptions separately.

Acknowledgments

We acknowledge the input of Prof. Aarne Ranta who created GF and Krasimir Angelov, both of Chalmers University, Sweden, Juliet Mutahi and Kimani Njogu.

References

- D. A. Burke and K. Johannisson. 2005. Translating Formal Software Specifications to Natural Language. A Grammar-Based Approach. In P. Blache, E. Stabler, J. Busquets and R. Moot (eds), *Logical Aspects of Computational Linguistics (LACL 2005)*, Springer LNAI 3402, 51-66.
- Olga Caprotti. 2006. WebALT! Deliver Mathematics Everywhere. In C. Crawford et al. (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference*. 2164-2168).
- Jeri L. Moxely. Semantic Structure of Swahili Noun classes. In I. Maddieson and T. Hinnebusch (eds), *Language History and Linguistic Description in Africa: Trends in African Linguistics (2)*, Africa World Press.
- M. Forsberg and A. Ranta. 2005. Functional Morphology: Tool Demonstration. *FSMNLP 2005*, Springer LNCS 4002, 304-305.
- Arvi Hurskainen. 1992. A two-level computer formalism for the analysis of Bantu morphology: An application to Swahili. *Nordic Journal of African Studies*, 1(1):87-122.
- Arvi Hurskainen. 1999. SALAMA: Swahili Language Manager. *Nordic Journal of African Studies*, 8(2):139-157.
- Peter Ljunglöf and Staffan Larsson. 2008. A grammar formalism for specifying ISU-based dialogue systems. In B. Nordström and A. Ranta (eds), *Advances in Natural Language Processing (GoTAL 2008)*, LNCS/LNAI 5221, Springer.
- Aarne Ranta. 2004. A Type-Theoretical Grammar Formalism. *The Journal of Functional Programming*, 14(2):145-189.
- Aarne Ranta. 2009. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(2).
- Wanjiku Ng'ang'a. 2006. Multilingual content development for eLearning in Africa. *eLearning Africa: 1st Pan-African Conference on ICT for Development, Education and Training*. Addis Ababa, Ethiopia.

A Memory-Based Approach to Kikamba Named Entity Recognition

Benson N. Kituku¹

Peter W. Wagacha¹

Guy De Pauw²

¹School of Computing & Informatics

²CLiPS - Computational Linguistics Group

University of Nairobi

University of Antwerp

Nairobi, Kenya

Antwerp, Belgium

nebsonkituku@yahoo.com

guy.depauw@ua.ac.be

waiganjo@uonbi.ac.ke

Abstract

This paper describes the development of a data-driven part-of-speech tagger and named entity recognizer for the resource-scarce Bantu language of Kikamba. A small web-mined corpus for Kikamba was manually annotated for both classification tasks and used as training material for a memory-based tagger. The encouraging experimental results show that basic language technology tools can be developed using limited amounts of data and state-of-the-art language-independent machine learning techniques.

1 Introduction

The issue of Named Entity Recognition was one of the four themes of the Sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996). Although the focus was on defense related articles then, there has been a tremendous increase in research efforts over the years for different domains and languages, as presented in Nadeau and Sekine (2007). Named entity recognition (henceforth *NER*) can be defined as the task of recognizing specific concepts in a text, such as proper names, organizations, locations and the like. Part-of-speech tagging (POS tagging) is often mentioned in the same breath as *NER*, as it is used as an essential pre-processing step to accurate *NER*. POS tagging can be defined as assigning morphosyntactic categories to words.

Kikamba (Kamba) is a Bantu language spoken by almost four million Kamba people in Kenya, according to the 2009 population & housing census (Oparany, 2010). Most of this population lives in

the Machakos and Kitui counties and a substantial number along the Embu, Taita Taveta and Tharaka boundaries. For a long time the Kamba people have preserved their culture through carving, especially at Wamunyu and also basketry (*kiondo*) and traditional dance (*kilumi*). The Akamba Culture Trust (ACT) formed in 2005, is crusading for the preservation of culture through written form in literature and research departments. Despite the efforts of the organization and the number of people speaking the language, Kikamba still lacks basic language technology resources and tools. Only recently a spell checker was developed at the School of Computing & Informatics of the University of Nairobi in Kenya.

This paper focuses on the development of a Named Entity Recognizer for Kikamba. Having a good *NER* system for this language is useful for a wide range of applications, such as event detection with an emphasis on map and phrase browsing, information retrieval and general data mining. Building a successful *NER* system cannot really be done without an accurate part-of-speech tagger, unfortunately not available for Kikamba. In this paper we will outline how a part-of-speech tagger and named-entity recognizer can be built with a minimum of human effort, using annotated corpora and language-independent, state-of-the-art machine learning techniques.

2 Related Research

A wide variety of languages have been examined in the context of named entity recognition (Nadeau and Sekine, 2007) and part-of-speech tagging, but very few sub-Saharan African languages have such

tools available to them. Part-of-speech Tagging has been investigated in the South African language context. A number of tag sets and preliminary systems are available for Setswana (van Rooy and Pretorius, 2003), Xhosa (Allwood et al., 2003), Northern Sotho (Prinsloo and Heid, 2005; Taljard and Bosch, 2005; de Schryver and De Pauw, 2007; Faaß, 2010). Outside of South Africa, POS tagging for Swahili has been extensively researched using finite-state-techniques (Hurskainen, 2004) and machine learning methods (De Pauw et al., 2006) and some preliminary experiments on Luo have been described in De Pauw et al. (2010).

Swahili is also - to the best of our knowledge - the only Bantu language that has been studied in the context of named-entity recognition (Shah et al., 2010). A few research efforts however investigate the problem of recognizing African named entities in regional varieties of English, such as South African English (newspaper articles) (Louis et al., 2006) and Ugandan English (legal texts) (Kitoogo et al., 2008).

3 Approaches in building the classifier

There are roughly two design options available when building a classifier for NER. The first one involves hand crafting a dictionary of names (*a gazetteer*) and an extensive list of hand-written disambiguation rules. This option is time consuming, particularly for a less-studied language such as Kikamba. Another option is to use techniques that learn the classification task from annotated data. This has the advantage that different techniques can be investigated for the same annotated corpus and that evaluation is possible by comparing the output of the classifier to that of the reference annotation (see Section 6).

As our machine learning algorithm of choice, we have opted for Memory-Based Learning (MBL) (Daelemans and van den Bosch, 2005). MBL is a lazy learning algorithm that simply takes the training data and stores it in memory. New data can be classified by comparing it to the items in memory and extrapolating the classification of the most similar item in the training data. For our experiments we used MBT (Daelemans et al., 2011a), which is a wrapper around the memory-based learning software TiMBL (Daelemans et al., 2011b) that facili-

tates the learning of sequential classification tasks.

4 Corpus Annotation

To build a machine-learning classifier for POS tagging and NER, an annotated corpus needs to be built. Previous research (de Schryver and De Pauw, 2007) showed that it is possible to quickly build a POS tagger from scratch on the basis of a fairly limited amount of data. The experiments described in this paper explore this train of thought for the Kikamba language and further extend it to the NER classification task.

A manually cleaned and automatically tokenized web-mined corpus of about 28,000 words was manually annotated for parts-of-speech and named entities. For the former annotation task a very small tag set was used that identifies the following parts of speech: `noun`, `verb`, `adverb`, `adjective`, `preposition`, `punctuation`, `interjection`, `cardinal`, `pronoun` and `conjunction`. These coarse-grained tags can be used in future annotation efforts as the basis for a more fine-grained tag set.

The NER annotation uses the IOB tagging scheme, originally coined by Ramshaw and Marcus (1995) for the natural language processing task of phrase chunking. The IOB scheme indicates whether a word is inside a particular named entity (I), at the beginning of an entity (B)¹ or outside of an entity (O). We distinguish between three types of named entities, namely persons (PER), organizations (ORG) and locations (LOC).

Since one of the main bottlenecks of NER for non-Indo-European languages is the lack of gazetteers of foreign names, we also added an additional 2000 Kikamba words for place, people and organization names. These were also used to facilitate and speed up the annotation process.

Manual annotation of the words was done using a spreadsheet. This manual process also helped to detect anomalies (and errors) which had not been resolved during the cleaning stage, hence improving the quality of the classifier. Each word was placed on separate row (token) with subsequent

¹In practice, the B tag is only used to mark the boundary between two immediately adjacent named entities and is therefore relatively rare.

Token	POS Tag	NER category
Ūsumbĩ	noun	I-ORG
wa	preposition	O
Ngai	noun	I-PER
nĩ	conjunction	O
kyaũ	adjective	O
?	punc	O

Table 1: Sample annotation of Kikamba corpus.

columns providing a drop-down box for parts of speech and named entity classes. A very small sample of the annotated corpus can be found in Table 1.

5 Features for Classification

During classification words are handled differently according to whether they are considered to be *known* or *unknown*. Known words are tokens that have been encountered in the training data and for which classification can usually be accurately done on the basis of local context by looking at the surrounding words and classes. For unknown words, i.e. words that have never been seen before, we also add pseudo-morphological information to the information source, such as the first and last n characters of the word to be tagged and information about hyphens and capitalization. Particularly the last feature is important, since during POS tagging the identification of (proper) nouns is paramount to the NER system.

The Memory-based Tagger (MBT) builds two separate classifiers for known and unknown words. The optimal set of features for classification was experimentally established. For known words, the best context considered two disambiguated part-of-speech tags to the left of the word to be tagged and one (not yet disambiguated) tag to the right. The accuracy of the part-of-speech tagger ($> 90\%$) can be found in Table 3.

For the unknown words group we included the aforementioned pseudo-morphological features alongside the typical contextual ones. We used a local disambiguation context of only one tag on both sides. Increasing the context size to be considered resulted in an increase in classification errors: the training data is apparently too limited to warrant a

larger context to generalize from when it comes to classifying unknown words. The average tagging accuracy of 71.93% (Table 3) shows that more data will be needed to arrive at a more accurate handling of unknown words.

In view of the morphological structure of the Kikamba language many words will start with Mb (e.g. Mbui - goat), Nd (Ndua - village), Ng (Ng’ombe - cow), Ny (Nyamu - wild animal), Th (Thoa - price), Mw (Mwaka - year), Kw (Kwangolya - place name), Ky (Kyeni - light), Ma (maiu - bananas), Sy (Syombua - person name). These examples show that even considering only the first two letters of unknown words is a good idea, as these are usually quite indicative of their morphosyntactic class, in this case the nominal class.

Furthermore, we also consider the last two letters, as most names of places in the Kikamba language will end in *-ni*, e.g. Kathiani, Kaviani, Nzaikoni, Makueni and Mitaboni. As mentioned before we also include capitalization as an import feature towards disambiguation. The hyphenation feature however did not provide much improvement in terms of accuracy. For the Kikamba language an interesting feature would indicate the presence of a single-quote (’) in the word, as this can also be an informative discriminating factor (e.g. for the words Ng’aa, Ng’ala, Ng’anga, Ng’eng’eta, Ng’ombe, Ng’ota etc.). In future work, we will investigate ways to introduce such language-specific orthographic features in the machine learning approach.

6 Experiments and Results

In this section we will describe the experimental results obtained on the basis of our small annotated corpus of 28,000 words. Various metrics were used for the evaluation of both the POS tagger and the NER system: accuracy, precision, recall and F-score. Accuracy simply expresses the number of times the classifier made the correct classification decision. Precision on the other hand calculates for each class how many times the class was correctly predicted, divided by the number of times that particular class was predicted in total. Recall on the other hand is defined as the number of times a class was correctly predicted, divided by the number of

Metric	POS tagging	NER
Precision	83.24	96.47
Recall	72.34	87.13
F-score	77.41	91.56

Table 2: Recall, precision and F-score for both classifiers

times that particular class appears in the test data. Finally, the F-score is the harmonic mean of recall and precision, calculated as outlined in Formula 1. The precision weighting factor β was set to 1.

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \quad (1)$$

6.1 Experimental Setup

The k-folds evaluation method (Weiss and Kulikowsie, 1991) was used to perform the evaluation of the system. This method was selected because of the relatively small size of the Kikamba corpus. We used a k value of 10. This means the annotated corpus is randomly portioned in ten equal folds (respecting sentence boundaries). For each experimental *fold*, one part was used as the evaluation set, while the other nine parts made up the training set. This ensures that evaluation takes place over all of the annotated data. The biggest advantage of this kind of experimental setup is that we can measure the accuracy on data that is not known to the system during training. By comparing the output of the classifiers to the annotation, we can calculate the aforementioned evaluation scores.

6.2 Results

Tables 2 and 3 outline the experimental results. The latter shows the accuracy for both classification tasks and for each of the ten partitions, followed by the average score. The former table shows precision, recall and F-score. These figures were obtained by calculating precision and recall for each class (i.e. each part-of-speech tag and named entity class) and averaging the scores.

Table 2 shows a precision of 83.24% and 96.47% for POS tagging and NER respectively. Error analysis showed that for the part of speech tagging task there was substantial false positive classification which lowered the percentage. A closer look at

the individual class categories for POS tagging and the confusion matrix extracted from MBT, indicates that the noun and preposition classes were particularly vulnerable to the effect of false positives. For the NER system the least false negatives were seen for class ‘‘O’’ with the other classes doing reasonably well too.

The recall scores in Table 2 are significantly lower than the precision scores. The low recall score for part-of-speech tagging is mainly due to a rather bad handling of verbs and numerals. The latter result is surprising since numerals should be straightforward to classify. More worryingly is the bad recall score for verbs. Future work will include an error analysis to identify the bottleneck in this case. The recall scores for NER on the other hand are more encouraging. The main bottleneck here is the handling of B- type tags. Most likely there is not enough data to handle these rather rare classes effectively.

Finally, the F-score stands at 77.41% and 91.56% for POS tagging and NER respectively. The F-score for POS tagging suffers because of the recall problem for verbs, but the F-score for NER is very encouraging, also compared to the results of Shah et al. (2010), who describe an alternative approach to NER for the Bantu language of Swahili.

Table 3 includes separate scores for known and unknown words. A closer look at the results for POS tagging indicates that, particularly given the very limited size of the training data, known words are actually handled pretty accurately (94.65%). Unknown words fare a lot worse at 71.93% accuracy. Error analysis shows that this is related to the aforementioned problem of verbal classification. A more well-rounded approach to modeling morphology within the classifier could provide a significant increase in unknown word tagging accuracy.

Compared to the results of the Swahili part-of-speech tagger described in De Pauw et al. (2006), the Kikamba system still has a long way to go. The Swahili tagger scored 98.46% and 91.61% for known and unknown words respectively with an overall performance of 98.25%. The Kikamba has an overall accuracy of 90.68%. This is obviously due to the difference in data set size: the Swahili corpus counted more than 3 million tokens, compared to only 28,000 words for the Kikamba tagger. Given this restriction, the performance of the tagger is sur-

FOLD	Part-of-Speech Tagging			Named Entity Recognition		
	Known	Unknown	Overall	Known	Unknown	Overall
1	94.24	78.01	92.07	98.81	98.44	98.76
2	94.59	73.65	90.64	98.13	88.07	96.21
3	94.55	71.31	90.22	98.60	93.00	97.56
4	94.79	68.44	90.75	98.67	94.68	98.07
5	93.44	71.43	89.74	98.77	91.28	97.48
6	94.34	68.62	90.41	98.76	97.56	98.58
7	95.85	67.05	90.43	99.33	95.73	98.64
8	95.46	70.25	91.06	98.81	95.96	98.31
9	95.42	83.64	92.69	98.59	90.63	96.75
10	93.80	66.86	88.74	99.13	95.39	98.42
Av.	94.65	71.93	90.68	98.76	94.07	97.88

Table 3: Experimental Results for the Part-of-Speech Tagging and Named Entity Recognition Tasks (10-fold cross-validation)

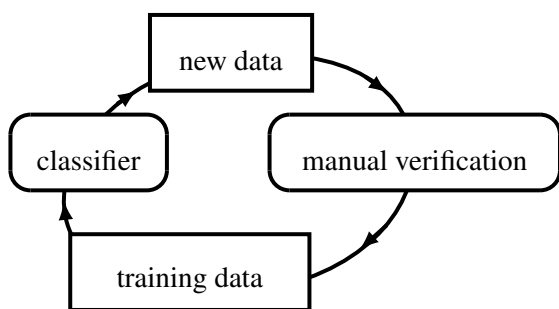


Figure 1: Semi-Automatic Annotation.

prisingly high.

For the NER task we report a performance of 98.76% and 94.07% for known and unknown words respectively, with an overall performance of 97.88%. Again, given the size of the data, this is an encouraging result and further evidence that data-driven approaches, i.e. techniques that learn a classification task on the basis of manually annotated data, are a viable way to unlock the language technology potentials of Bantu languages.

7 Conclusion and Future Work

We have presented a Kikamba Named Entity Recognizer with a classification accuracy of 97.88% and an F-score of 91.71% and a part-of-speech tagger with an accuracy of 90.68%. While the amount of

data is rather limited and we have not yet performed a full exploration of all experimental parameters, these scores are encouraging and further underline the viability of the data-driven paradigm in the context of African language technology.

We will investigate other machine learning techniques for these classification tasks and this data. As soon as a critical mass of training data is available, we will also perform *learning curve* experiments to determine how much data is needed to arrive at accuracy scores comparable to the state-of-the-art in NER and POS tagging.

At this point, we can use the systems described in this paper, to semi-automatically annotate larger quantities of data. This process is illustrated in Figure 1: we use the currently available training data to train a classifier that automatically annotates new data. This is then checked manually and corrected where necessary. The resulting data can then be added to the training data and a new classifier is trained, after which the cycle continues. This type of semi-automatic annotation significantly improves the speed and consistency with which data can be annotated. As such the systems described in this paper should be considered as the first bootstrap towards an expansive annotated corpus for Kikamba.

References

- J. Allwood, L. Grönqvist, and A. P. Hendrikse. 2003. Developing a tagset and tagger for the African lan-

- guages of South Africa with special reference to Xhosa. *Southern African Linguistics and Applied Language Studies*, 21(4):223–237.
- W. Daelemans and A. van den Bosch. 2005. *Memory-based Language Processing. Studies in Natural Language Processing*. Cambridge University Press, Cambridge, UK.
- W. Daelemans, J. Zavrel, A. van den Bosch, and K. van der Sloot. 2011a. MBT: Memory Based Tagger, version 3.2, Reference Guide. ILK Research Group Technical Report Series 10-04, Tilburg.
- W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2011b. TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide. ILK Research Group Technical Report Series no. 10-01 04-02, Tilburg University.
- G. De Pauw, G-M de Schryver, and P.W. Wagacha. 2006. Data-driven part-of-speech tagging of Kiswahili. In P. Sojka, I. Kopeček, and K. Pala, editors, *Proceedings of Text, Speech and Dialogue, Ninth International Conference*, volume 4188/2006 of *Lecture Notes in Computer Science*, pages 197–204, Berlin, Germany. Springer Verlag.
- G. De Pauw, N.J.A. Maajabu, and P.W. Wagacha. 2010. A knowledge-light approach to Luo machine translation and part-of-speech tagging. In G. De Pauw, H. Groenewald, and G-M de Schryver, editors, *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pages 15–20, Valetta, Malta. European Language Resources Association (ELRA).
- G-M. de Schryver and G. De Pauw. 2007. Dictionary writing system (DWS) + corpus query package (CQP): The case of Tshwanelex. *Lexikos*, 17:226–246.
- G. Faaß. 2010. The verbal phrase of Northern Sotho: A morpho-syntactic perspective. In G. De Pauw, H.J. Groenewald, and G-M. de Schryver, editors, *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pages 37–42, Valletta, Malta. European Language Resources Association (ELRA).
- R. Grishman and B. Sundheim. 1996. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.
- A. Hurskainen. 2004. *HCS 2004 – Helsinki Corpus of Swahili*. Compilers: Institute for Asian and African Studies (University of Helsinki) and CSC.
- F. Kitoogo, V. Baryamureeba, and G. De Pauw. 2008. Towards domain independent named entity recognition. In *Strengthening the Role of ICT in Development*, pages 38–49, Kampala, Uganda. Fountain Publishers.
- A. Louis, A. De Waal, and C. Venter. 2006. Named entity recognition in a South African context. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, SAICSIT '06, pages 170–179, Republic of South Africa. South African Institute for Computer Scientists and Information Technologists.
- D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January.
- W.A. Oparany. 2010. *2009 Population & Housing Census Results*. Available from [[http://www.knbs.or.ke/Census_Results/Presentation by Minister for Planning revised.pdf](http://www.knbs.or.ke/Census_Results/Presentation_by_Minister_for_Planning_revised.pdf)], Nairobi, Kenya.
- D. J. Prinsloo and U. Heid. 2005. Creating word class tagged corpora for Northern Sotho by linguistically informed bootstrapping. In *Proceedings of the Conference on Lesser Used Languages & Computer Linguistics (LULCL-2005)*, Bozen/Bolzano, Italy.
- L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.
- R. Shah, B. Lin, A. Gershman, and R. Frederking. 2010. Synergy: A named entity recognition system for resource-scarce languages such as Swahili using online machine translation. In G. De Pauw, H.J. Groenewald, and G-M de Schryver, editors, *Proceedings of the Second Workshop on African Language Technology (AfLaT 2010)*, pages 21–26, Valletta, Malta. European Language Resources Association (ELRA).
- E. Taljard and S. E. Bosch. 2005. A comparison of approaches towards word class tagging: disjunctively vs conjunctively written Bantu languages. In *Proceedings of the Conference on Lesser Used Languages & Computer Linguistics (LULCL-2005)*, Bozen/Bolzano, Italy.
- B. van Rooy and R. Pretorius. 2003. A word-class tagset for Setswana. *Southern African Linguistics and Applied Language Studies*, 21(4):203–222.
- S.M. Weiss and C.A. Kulikowsie. 1991. *Computer systems that learn*. Morgan Kaufmann, San Mateo, CA, USA.

Morphological Analysis of Gikūyū using a Finite State Machine

¹Kamau Chege ¹Wanjiku Ng'ang'a
¹Peter W. Wagacha

¹School of Computing and Informatics, University of Nairobi, Kenya.
kamauchege@gmail.com, wanjiku.nganga@uonbi.ac.ke, waiganjo@uonbi.ac.ke

²Guy De Pauw, ³Jayne Mutiga

²CLiPS - Computational Linguistics Group, University of Antwerp, Belgium

³Centre for Translation and Interpretation, University of Nairobi, Kenya.

guy.depauw@ua.ac.be,
jaynemutiga@yahoo.co.uk

Abstract

In this paper we present the development of a morphological analysis system for Gikūyū. Major morphological processes prevalent in Gikūyū language are explored. These include verb inflection, verb and noun derivation and verb reduplication. In this work, finite state transducers are used to model Gikūyū morphology. Xerox finite state tools are used to build the lexical transducers for the system and to model rewrite rules prevalent in Gikūyū.

The system achieves an acceptable representation of Gikūyū morphology. It can correctly analyze Gikūyū words with an accuracy of 56%. As Gikūyū is highly inflections, ambiguity is a big challenge both in morphological analysis and machine translation.

1 Introduction

Morphological analysis is an important first step in many natural language processing tasks such as parsing, machine translation, information retrieval, part-of-speech tagging among others. The field of natural language processing has progressed considerably over recent years. Despite this, many African languages have been left behind. This status quo has been influenced by two major reasons. Firstly, many African countries have adopted European international languages such as English and French among others as their official languages. This makes local languages unpopular and therefore not economically viable to invest in. Secondly, many African languages are resource-scarce. There is very little, if any, digitized linguistic resources for these languages. Moreover, lack of financial resources and political will hinders creation of such linguistic resources from scratch. Gikūyū is one of the many

resource-scarce African languages. Rapid revolution in Information and Communication Technology is affecting the way we learn, socialize, and do business among other things. There is need to position our local languages in this new paradigm else they will be lost. This can only be achieved through availing digital resources for them; and linguistic tools are vital in creation and dissemination of such resources. Gikūyū is a highly agglutinative Bantu language spoken by between 6-7 million speakers in Central Kenya.

2 Literature Review

2.1 Previous work

A number of research initiatives have gone into morphological analysis of African languages. Various approaches have been taken on morphological analysis. De Pauw and Wagacha (2007) use unsupervised methods to learn Gikūyū morphology. In their work, maximum entropy learning is used to carry out automatic induction of shallow morphological features for Gikūyū. This approach is desirable given that it achieves morphology with minimal human effort, but availability of sufficient corpus to facilitate learning is a challenge.

Other NLP research works on Gikūyū language include automatic diacritic correction using grapheme-based memory model (De Pauw et.al, 2007) and a Gikūyū text-to-speech system using Festival tool. On Swahili, a Bantu language closely related to Gikūyū, an attempt at morphological analysis based on two-level morphology (Koskeniemmi, 1983) is carried out in a project named SWATWOL (Hurskainen, 2004) at the University of Helsinki, Finland. In this work,

Xerox's TWOLC tool is used to implement two level rules.

Other research works at morphological analysis of African languages include Kikamba Morphological analyzer using rules, SwaMorph analyzer for Swahili using finite state technology among other efforts on Central and Southern African languages.

2.2 Gikūyū Morphology

Gikūyū is a language spoken by a Kenyan community of Kamba-Kikuyu subgroup of the Bantu origin, with approximately 7 million speakers living in Central Kenya. The language has six dialects and is lexically similar to closely related languages such as Chuka, Embu, Meru, Kamba.

Gikūyū is a highly inflectional language with a complex word structure and phonemics. Like many other Bantu languages, Gikūyū has 15 noun classes and two additional locative classes. The language also has a concord system formed around the noun classes. The language is also tonal, a major source of ambiguity.

Noun Morphology: Gikūyū nouns can be grouped into two categories namely derived and underived nouns. Underived nouns consist of named entities. Derived nouns can be formed in two ways. Firstly, they are formed by affixing prefixes of diminutive, augmentative or collective to an underived noun. Examples are,

thitima -gĩ-thitima (a big light bulb)
mũndũ – ma-mũndũ (many big persons)
i-ndathi – rũ-ndathi (a bad/ugly) bun)

The second form of derived nouns is formed through nominalization of verbs. This involves circumfixing verb roots with a set of prefix and suffixes to give different meaning depending on the type of nominalization. Nominalization types include agentive (most prevalent), occasion, manner, locative e.t.c. Examples include,

mũ-thaak-i (player) i-ho-ero (Place of prayer)
i-ge-th-a (harvesting occasion) ga-thom-i (the small one who reads)

The membership of a noun to a noun class is determined slightly by its initial characters but is mainly determined by the concord system which it enforces on other parts of speech in a sentence. All Gikūyū nouns, underived or otherwise, can be affixed with a suffix *-inĩ* with the effect of

changing the meaning from a referential entity to a location.

Verb Morphology: Gikūyū language is agglutinative. Dependent morphemes are affixed to the independent morpheme to derive a certain meaning to the surface verb. A typical Gikūyū verb consists of a combination zero or more of the dependent morphemes and a mandatory dependent morpheme, with the final vowel also being mandatory. Figure 1 illustrates the verb morphology.

The simplest verb consists of the verb root and the final vowel. These are usually commands or directives. Subjunctive verb formations i.e. commands, can optionally take a plural marker *-i* or *-ni*. Examples of Verbs are shown below;

ma-thom-e (so that they read)
ci-ti-raa-tũ-meny-ag-a (they were not knowing us)
a-gaa-kenaken-ithi-ag-io (he/she will be made happy a little more)
nĩ-kĩ-mũ-hũr-ag-a (it usually beats him/her)
nd-aa-ngĩ-kaa-ma-caracar-ithi-ang-ia (he would not have help them a little more in searching)
kom-a-i (sleep)
rehe-ni (bring)

Reduplication: Gikūyū verbs also undergo verb reduplication. This involves repeating part or the entire lexical root of the verb, depending on the number of syllables in the root. The meaning: *Focus+ Subj_Marker+ Neg+ Cond+ Tense+ Obj_Marker+ Redupl+ Verb+ Dev_Ext+ Aspect+ FV* derived from reduplication varies among verb stems but usually means repeatedly doing the action, doing the action for a little longer, among others.

Verbs with one or two syllables undergo full reduplication. Verbs with more than two syllables undergo partial reduplication. Only the first two syllables are repeated. In addition, the last vowel, whatever character it is, is rewritten as 'a'. Examples are;

koma - koma-koma (sleep a little) ne – nea-nea (give a little)
negen-a – nega-negen-a (make noise a little more)
tiga – tiga-tiga (leave a little)
hungura – hunga-hungura (slip under a little more)

Verb Mutation: Gikūyū verbs are also affected by consonantal and vowel phonemics.

Prenasalized stop formation (and its variant called Meinhof’s Law) involves consonants *b, c, r, t, g, k* being replaced with NC composites in verbs obeying first person singular, noun classes 1, 8 and 9 concord systems. Vowel-initial verbs whose first consonant is any of the participating consonants also undergo this mutation. Examples include;

roota – ndoota ūma -nyūmia
guna – ng’una komia – ngomia
cuuka – njuuka egeka njegeka

Dahl’s Law is a consonantal mutation that involves the voiceless trigger sound *-k* appearing before other voiceless sounds *c,k,t* being replaced with its equivalent voiced sound *-g-*. Examples include;

kū-thoma – gūthoma ka-ka-thaaka – gagathaaka
ma-kaa-ka-menya - magaakamenya

Vowel mutation includes vowel lengthening before prenasalized stops and vowel assimilation when some vowel combinations appear in the neighborhood of each other.

2.3 Finite State Transducers

The ability to formally describe language lexicon and morphological rewrite rules using finite state machines have made it a popular approach to computational morphology. The two-level formalism has been successfully described using finite state transducers. Finite state networks model grammar rules which can be used to validate if a given input string belongs to a language.

Finite state transducers are used to model both morphological generators and analyzers. An FST generator uses rules to transform a lexical form into a surface form. An FST analyzer consults both rules and the lexicon to transform a surface string into the corresponding lexical form. See Figure 2.

Since FSTs are bidirectional, the two processes are usually the opposite of each other. Finite state transducers can also be used to carry out other NLP tasks such as tokenization, part-of-speech tagging among others.

2.4 Two-Level formalism

Transformation of input strings from lexical to surface representation involves a series of intermediate stages. Transitions between these stages involve zero or more generative rules called rewrite rules. Classical generative phonology implements rewrite rules in sequence. Koskeniemmi (1983) two-level formalism allows the two representations to be directly related with no need for intermediate steps. The formalism models the relation between the two formalism using parallel rules. The rules act a set of conditions, not actions, whose role is to ascertain whether the correspondence between the two representations is correct.



Figure 3: Two Level Representation

This formalism is bidirectional, which allows generation and analysis to be modeled in parallel. It can also be easily represented using finite state transducers.

2.5 Challenges

Gikūyū language is more spoken than written (and read). This means that very few written sources exist for the language. Furthermore, the language use is losing favor to the more popular national languages; Swahili and English.

The Gikūyū orthography includes two diacritically marked characters *ĩ* and *ũ* that represent different phonemes from *i* and *u* respectively. Standard keyboard lacks these two characters and hence many users tend to use their unmarked equivalents. In addition, the characters complicate automated corpus collection through such methods as OCR.

Gikūyū language is also tonal; this is usually a major source of word ambiguity.

3 Methodology

3.1 Corpus Collection

This work uses a 25,000 word corpus, corrected from a variety of resources. Out of this, a set 19,000 words is from previous works (De Pauw and Wagacha, 2007; De Pauw et.al, 2007) carried out on Gikūyū language. The set has a bias religious material but also includes text from hymn books, poems, short stories, novels and

also web crawling. A small set is also manually transcribed. The remaining set of 6,000 words is collected during this project and includes materials from manual transcription, constitutional review, online religious material, agricultural material, blog entries among others. It is mainly running text and part of it is held out as test data.

The data is cleaned and annotated manually and also automated by writing perl scripts. This involved removing non-Gīkūyū words, part-of-speech tagging and also grouping using structure similarity.

3.1 Xerox Tools

Xerox's Finite state tools, XFST and LEXC, were used to implement the Gīkūyū morphology system. Due to an encoding problem appearing during compile-replace for reduplication, the two diacritically marked characters are represented by two characters not in Gīkūyū alphabet. S represents ũ while L represents ĩ. The Gīkūyū lexicon files were developed using Xerox's lexicon format so as to be compiled using the LEXC compiler. The files were modularized along the major parts of speech.

3.2 Morphology System

The morphology system is modeled around Xerox's LEXC tool for creating large lexicon. Morphological markers are used to guide the meaning of each morpheme. Reduplication and rewrite rules are modeled using XFST tool and then composed with the lexicon after it has been compiled. The lexicon is implemented as continuation classes and organized around various parts of speech. Underived nouns, together with their diminutives, augmentatives, collectives and the locative modifier are implemented as a single lexicon as shown in figure 4.

Derived nouns are implemented in a separate lexicon file. To enforce circumfixation (prefix-suffix combination) associated with nominalization, flag diacritics are used, Gīkūyū verbs have a number of optional affixes. These are implemented through allowing a -i- transition in every continuation class that is optional. To enhance the organization of the verb lexicon file, verb roots are placed in different continuation classes depending on how structurally similar they are, as shown below.

LEXICON SPList (Have irregular endings)
he Suff;

ne Suff;

LEXICON IAMidLowList (ends with -ia, obeys mid-low sound)

endia: end Suff;
reki: rek Suff;

LEXICON IAMidHighList (ends with -ia, obeys mid-high sound)

akia: ak Suff;
gira: gir Suff;

LEXICON ACausMidLowList (ends with -a and are causative, obeys mid-low sound)

cera: cer Suff;
gema: gem Suff;

LEXICON ACausMidHighList (ends with -a and are causative, obeys mid-high sound)

baca: bac Suff;
gwata: gwat Suff;

LEXICON AMidLowList (ends with -a, are not causative, obeys mid-low sound)

kombora: kombor Suff;
hehenja: hehenj Suff;

LEXICON AMidHighList (ends with -a, are not causative, obeys mid-high sound)

aka: ak Suff;
kanya: kany Suff;

Reduplication is implemented using compile-replace function in XFST tool. This is a 2-stage process. The first stage involves using regular expressions to identify sections of the verb to be reduplicated and replacing them with regular expressions as shown,

1. define *Redup1* *[[Syllables]^2]* @-> “*^ [{ ... Z } ^2] Z*” *||*+*[Redup]*“*[EFLAGS]** *_[Alpha]+* *+ [Verb]*”];
2. define *Redup2* *[Syllables CON^{0,1}]* @-> “*^ [{ ... Q } ^2 ^] Q*” *||*+*[Redup]*“*[EFLAGS]** *_[+ [Verb]]*”];
3. define *Redup3* *[CON]* @-> “*^ [{ ... V } ^2 ^] V*” *||*+*[Redup]*“*[EFLAGS]** *_[+ [Verb]]*”];
4. define *Reduplication* *Redup1 .o. Redup2 .o. Redup3*;

The compile-replace command is then invoked.

Rewrite rules for several aspects are also implemented using XFST. These include Dahl's Law, prenasalization, vowel assimilation, prenasals removal by diminutives, augmentatives and collectives among others.

1. define *DahlsLaw* [k -> g || _ [[a|e|i|ĩ|o|u|ũ] [FLAGS]* [[a|e|i|ĩ|o|u|ũ]* [[c|k|t][FLAGS] [a|e|i|ĩ|o|u|ũ]*^>0 [FLAGS]* [c|k|t]]];
2. define *PrenasalAugmentative* {nd} -> t
|"+[Noun]" [ALLFLAGS]* ["+[Dim]" k a
|"+[Augm]" k L | "+[Augm_Derog]" r S |
|"+[Augm_Coll]" m a | "+[Dim_Coll]" t
S][ALLFLAGS]* "+[9NC]" [ALLFLAGS]* _;

Other parts of speech implemented include adjectives, possessives, demonstratives, associatives, conjunctions, and prepositions. It is important to note that all but conjunctions and prepositions follow the concord system determined by the nouns they describe. Several analysis examples are shown below,

Nĩmakaamathaambagia

1. +[Focus]+ [3P_PL]+ [Subj]+ [Rem_Future]+ [2NC]+ [Obj]*thaamba*+ [Verb]+ [De-vExt_Causative]+ [Aspect_Habit]+ [FV_Ind]
2. +[Focus]+ [3P_PL]+ [Subj]+ [Rem_Future]+ [6NC]+ [Obj]*thaamba*+ [Verb]+ [De-vExt_Causative]+ [Aspect_Habit]+ [FV_Ind]
3. +[Focus]+ [2NC]+ [Subj]+ [Rem_Future]+ [2NC]+ [Obj]*thaamba*+ [Verb]+ [De-vExt_Causative]+ [Aspect_Habit]+ [FV_Ind]
4. +[Focus]+ [2NC]+ [Subj]+ [Rem_Future]+ [6NC]+ [Obj]*thaamba*+ [Verb]+ [De-vExt_Causative]+ [Aspect_Habit]+ [FV_Ind]
5. +[Focus]+ [2NC]+ [Subj]+ [Rem_Future]+ [3P_PL]+ [Obj]*thaamba*+ [Verb]+ [De-vExt_Causative]+ [Aspect_Habit]+ [FV_Ind]

kimagai

1. *kima*+ [Verb]+ [Aspect_Habit]+ [FV_Ind]+ [Subjun_PL]

gakaari

1. +[Noun]+ [Dim]+ [9NC]*ngaari*

magoondu

1. +[Noun]+ [Augm_Coll]+ [9NC]*ng'ooundu*

tũguuka

1. +[Noun]+[Dim_Coll]+[1NC]*guuka*

nyũmba

1. +[Noun]+[9NC]*nyũmba*
2. +[Noun]+[10NC]*nyũmba*

Word generation is carried much the same way as analysis, but in the reverse, as shown below;

+ [Noun]+[10NC]*mwana*
1. *ciana*

4 Testing

In testing the morphology system we encounter a challenge as Gĩkũyũ, being a resource scarce language, has no existing standards against which this work can be evaluated against. We therefore adopt quantitative evaluation of the system's performance.

4.1 Morphology System

The system is evaluated on its efficiency in recognition of Gĩkũyũ words and correctness of analysis of test data. 100 words were randomly picked from the test data and analyzed using the morphological analysis system. The possible outputs from the evaluation point of view were;

- i) Recognized and correctly analyzed,
- ii) Recognized and incorrectly analyzed, and
- iii) Not recognized.

5 Results

From the tests, it was observed that non-recognized words were mainly caused by the root form not being included in the lexicon files. Another category of non-recognized words was caused by the writers influence on spelling especially on vowel length and assimilation.

Result	Correct Analysis	Incorrect Analysis	Not recognized	
No. of instances	56	7	37	100
Precision	56/62 = 89.9%			
Recall	56/88 = 63.64%			
Success rate	56/100=56%			

Table 1: Morphology Results

6 Conclusion

In this work, we have explored the use of finite state methods to model morphological analysis of Gĩkũyũ, a resource-scarce Bantu language. Since Gĩkũyũ language is closely related to a number of Bantu languages, we propose the use of this knowledge and tools be applied to development of such languages.

Future work includes the application of the developed morphology system to implement a proof-of-principle shallow-transfer machine translation system for Gĩkũyũ to English.

Acknowledgments

The research work presented in this paper was made possible through the support provided by

Acacia Programme in IDRC, Canada through the PALnet/ANLoc Project.

Focus+Subj_Marker+Neg+Cond+Tense+Obj_Marker+Redupl+Verb+Dev_Ext+Aspect+FV

Figure 1: Verb Morphology

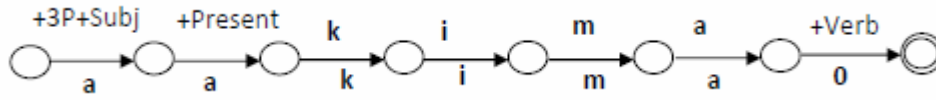


Figure 2: A Transducer Example

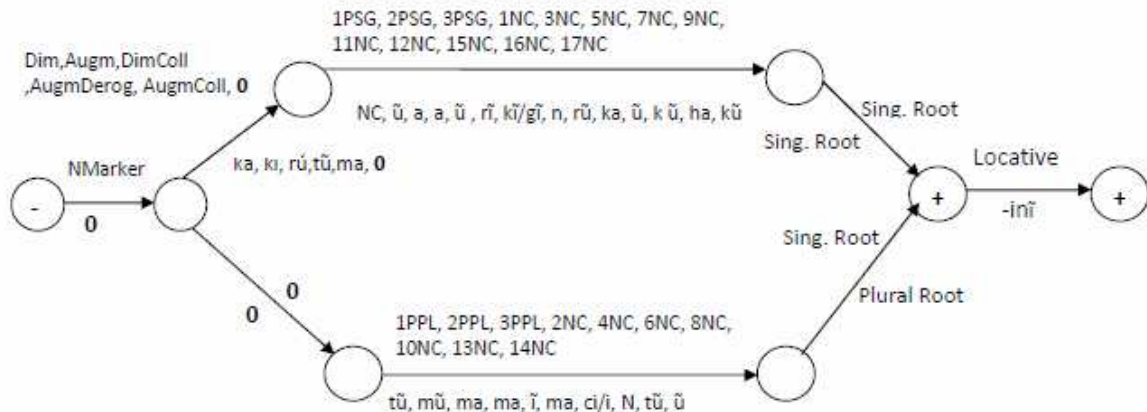


Figure 4: Underived Nouns

References

Beesley K. R. and Kartunen L., 2003, Finite-State Morphology, *CSLI Publications*, Stanford.

Dalrymple M., Liakata M., Mackie L., 2006, Tokenization and Morphological analysis of Malagasy, *Computational Linguistics and Chinese Language Processing*, Association for Computational Linguistics and Chinese Language Processing

G. De Pauw, Wagacha P., 2007, "Bootstrapping Morphological Analysis of Gĩkũyũ Using Unsupervised Maximum Entropy Learning". *Proceedings Eighth INTERSPEECH Conference*.

G. De Pauw, Wagacha P., De Schryver G. 2007, "Automatic Diacritic Restoration for Resource Scarce Languages". *Proceedings of Text, Speech and Dialogue, Tenth International Conference Heidelberg, Germany*.

<http://www.aflat.org/?q=biblio> *Publications on Natural Language Processing research Papers on African Languages*. (Accessed 12th March 2011)

Hurskainen A., 2004, HCS 2004 - Helsinki Corpus of Swahili. *Compilers: Institute for Asian and African Studies (University of Helsinki) and CSC – Scientific Computing*.

Inaki A. et al, 2007, Shallow Transfer MT Engine for Romance Languages in Spain, *Universitat d'Alacant*.

Koskeniemmi K., 1983, Two-Level Morphology: A new Computational Model for Word-Form Recognition and Production, *University of Helsinki*.

Mugane J., 1997, A Paradigmatic Grammar of Gĩkũyũ *CSLI Publications*, Stanford California.

Understanding Natural Language through the UNL Grammar Workbench

Sameh Alansary
Bibliotheca Alexandrina,
Alexandria, Egypt
&
University of Alexandria,
Faculty of Arts, Department
of Phonetics and Linguistics,
Alexandria, Egypt.
Sa-
meh.alansary@bibalex
.org

Magdy Nagi
Bibliotheca Alexandrina,
Alexandria, Egypt
&
University of Alexandria,
Faculty of Engineering,
Dept. of Computer and Sys-
tem Engineering, Alexandria,
Egypt
Mag-
dy.nagi@bibalex.org

Noha Adly
Bibliotheca Alexandrina,
Alexandria, Egypt
&
University of Alexandria,
Faculty of Engineering,
Dept. of Computer and System
Engineering, Alexandria,
Egypt.
No-
ha.adly@bibalex.org

Abstract

This paper is an attempt to introduce the Universal Networking Language (UNL) as a tool for achieving natural language understanding through a capable grammatical framework. The UNL system utilizes a processing workbench capable of effectively and accurately extracting the universal meaning behind the sentences of any language and, thus, analyzing and generating natural language words, sentences and texts. Such a framework can subsequently be used by linguists in the field of natural language processing (NLP) for applications such as machine translation (MT), question answering, information retrieval, etc.

1 Introduction

The field of natural language processing was at some point in time referred to as Natural Language Understanding (NLU). However, today, it is well agreed that NLU represents the ultimate goal of NLP. Yet, that goal has not yet been accomplished as a full NLU System should be able to: a) paraphrase an input text; b) translate the text into another language; c) answer questions about the contents of the text; d) draw inferences from the text (Liddy, 2001)

Thus, the UNL system attempts to fulfill all of the previous criteria by meticulously and accurately analyzing the input text into a universal

abstraction of meaning. This meaning is represented in the form of a semantic network (the UNL network, or UNL expression). In this network, concepts are represented language-independently in the form of nodes, and each node is augmented with a wealth of semantic, grammatical and pragmatic information.

The grammatical foundation of the UNL system, thus, draws upon this information to determine the pure semantic relation between nodes. By determining them, the UNL can be said to have understood the natural language sentence; it can paraphrase this network into the same or other language, it can deduce certain information from its contents, etc. Moreover, using its robust grammars, the UNL system can generate a new meaning altogether and then generate it as a natural language sentence, in any language chosen.

The UNL grammar framework mainly adopts the X-bar theory as a foundation. The X-bar theory is in many respects similar to the UNL approach to natural language understanding. It assumes binary relations between the sentence constituents, which facilitates the process of mapping syntax onto semantics and vice versa. The X-bar theory also allows for many intermediate levels, a fact that gives the UNL great flexibility in the formation and decomposition of deep and surface syntactic structures.

In this paper, section 2 will start by examining the process of analyzing a natural language sentence. The process involves determining the exact meaning of words and the abstract relations they hold together in addition to encoding the other semantic, grammatical and pragmatic information they carry. In section 3, on the other

hand, the process of natural language generation is discussed. The capabilities of the generation grammar become clear in the way it is able to generate the constituent words in the target language, arrange them grammatically and make the necessary changes to the form of the word.

It is worth noting here that the arrangement of the following sections does not reflect the workflow of the UNL system or the ordered stages through which a natural language passes until it is finally transformed into a UNL semantic network, or vice versa. All of the following processes whether in analysis or generation work in unison and simultaneously to reach the most accurate understanding possible.

2 Analyzing Language

In order to claim the ability to fully and accurately understand human languages, a system must have the tools and methods capable of effectively decomposing the sentence into its basic constituents, understanding and encoding in some formal manner the intended meaning behind each constituent and the meaning reflected by its superficial grammatical form as well as its position in the sentence. It should also understand and encode the semantic relation between each constituent and the others.

The following subsections will present the techniques adopted by the UNL system to carry out the above processes; first, how a word in a natural language sentence is decomposed, analyzed and assigned the labels capable of bringing about a coherent and consistent sentence; second, how these words are linked together to form a syntactic structure then a semantic network that reflects the meaning of the whole sentence.

2.1 Analyzing Words

Words are the main conveyors of meaning in a sentence. The morphemes constructing a sentence carry the information without which a natural language sentence would be incomprehensible. “A positive absolute universal is that the morphemes of every language are divided into two subsystems, the open-class, or lexical, and the closed-class, or grammatical (see Talmy, 2000a). Open classes commonly include the roots of nouns, verbs, adjectives and adverbs and contribute most of the content. Closed classes, on the other hand, determine most of the structure and have relatively few members. They include bound forms such as inflections, derivations, and clitics; and such free forms as prepositions, con-

junctions, and determiners (Talmy, 2000) (Francis, 2005).

Encoding Functional and Grammatical Morphemes

To understand the full meaning of a sentence, closed classes must be acknowledged as they contribute to the meaning by cross-referencing the main concepts and indicating other linguistic and extra-linguistic information. Due to the semantic constraints on the closed-class subsystem, they constitute an approximately limited inventory from which each language draws in a unique pattern to represent its particular set of grammatically expressed meanings (Francis, 2005). This inventory is mimicked in the UNL system by a set of tags capable of representing the grammatical, semantic, pragmatic information that might be conveyed by the closed-class morphemes in any language (Alansary et al., 2010)¹.

Closed classes may be either represented as bound morphemes or free morphemes; bound morphemes are usually the result of inflection or derivation processes. The Arabic language, for example, is highly inflectional and is especially rich in word forms. Thus, Arabic word such as فتجاهلوني fatagaahaluunii ‘So they ignored me’, although a single orthographic word in Arabic, it is the equivalent of a whole phrase in some other languages. Therefore, in order to understand the full meaning of such a complex word, the information communicated by the bound morphemes in it must be included into its meaning.

Uncovering the bound morphemes in a word (i.e. affixes) and what they represent involves separating them from the core open-class concept by scanning the input words and matching them with the entries in the natural language-UNL dictionary; the longest most appropriate string match is chosen. However, there are usually several matches and, consequently, several potential analyses for a single input word. For example, figures 1 and 2 show two of the potential morphological analyses for the previous example word فتجاهلوني.

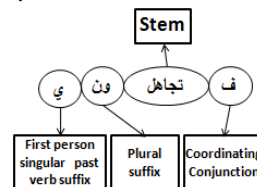


Figure 1. The first possible analysis for فتجاهلوني

¹ This set of tags and information about each is available at <http://www.unlweb.net/wiki/index.php/Attributes>

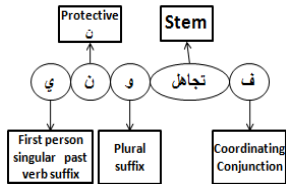


Figure 2. The second possible analysis for تجاهلوني

To resolve this sort of ambiguity, morphological disambiguation rules are used. Disambiguation rules assign priorities to the possible sequences of morphemes. Priorities range from 0 (impossible) to 255 (possible). In this case, the rules in (1) apply.

- (1a) (^PRS, ^PREFIX)(VER)(“ون”):=0;
 (1b) (V,PAST)(“و”, VSUFFIX,SPRON):=255;

Rule (1a) disqualifies the first analysis (figure 1) by stating that a past verb (not preceded by any of the present tense prefixes) can never have a “ون” as a suffix. On the other hand, rule (1b) maximally approves the string “و” being a suffix for a past verb. In the same manner, all of the constituent morphemes are disambiguated and the wrong analyses are refuted until only the most appropriate analysis is left which is the analysis in (figure 2).

In addition to bound morphemes, other closed-class members are free forms such as conjunctions, prepositions, auxiliaries, etc. These are also scanned and matched with dictionary entries; however, these morphemes are encoded using special tags into the final semantic network². For example, conjunctions such as بعد baEd ‘after’ and إذا ithaa ‘if’ are replaced by the tags “@after” and “@if” respectively. While adpositions like فوق fawq ‘above’ and حتى hattaa ‘until’ are replaced by “@above” and “@until” respectively.

Encoding Main Concepts

Aside from the previous grammatical or functional morphemes, the main content conveyed by a word is carried by a nominal, verbal, adjectival or adverbial lexical item that belongs to the open-class. After abstracting away all the functional bound morphemes from a word, the stem representing the main concept is left behind.

It is claimed that any of the concepts a person can know ought to have the potential to be expressed in any human language and that the se-

² These tags representing these too are found at <http://www.unlweb.net/wiki/index.php/Attributes>

semantic representations of words would be a particular type of concept (Francis, 2005). Thus, the UNL system has taken up a sort of language-independent conceptual representation to replace the members of the open-class words. This representation is a machine-readable numerical ID that stands for the exact sense the natural language word usually means. This ID is, in fact, adopted from the English WordNet 3.0. In the WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct unique sense (Fellbaum, 1998). The UNL system, then, uses the ID given to each synset in the WordNet ontology to refer to the universal concept it denotes. The use of the WordNet in building the UNL dictionaries and ontologies is discussed in (Boguslavsky et al., 2008) (Bekios et al., 2007), (Martins and Avetisyan, 2009), (Boudhh and Bhattacharyya, 2009).

For example, when encountering the Arabic word فتجاهلوني tagaahloni, the detected stem تجاهل tagaahala will be matched with the entries in the main natural language-UNL dictionary to determine the universal concept it stands for. However, this stem is also prone to some sort of ambiguity; it can either represent a noun meaning ‘ignoring’ or a verb meaning ‘ignore’. To determine which interpretation is intended here, lexical disambiguation rules come into effect; the rule in (2) resolves this ambiguity.

- (2) (NOU,MASDR)(“و”“ن”):=0;

This rule rules out the possibility of “تجاهل” being a noun since the possible noun alternative is a verbal noun and a verbal Arabic noun can never have a “و” or a “ن” as suffixes. Thus, “تجاهل” is established as a verb. However, even as a verb, there are five alternative senses to choose from³. Nonetheless, it can be argued at this point that the word is indeed understood in the sense that only the most appropriate interpretations are listed, all of which would be correct in different contexts and under different circumstances. For the purposes of our discussion, it will be presumed that this word is equivalent to the

³ The process of choosing the exact ID to represent the intended sense of the natural language lexical item is not an easy process. Apart from grammar rules, the UNL system makes use of extensive ontologies and knowledge bases that aid the process of word-sense disambiguation. However, this paper will only focus on the grammar-related solutions to word-sense disambiguation; the other techniques will be thoroughly discussed in forthcoming publications.

universal representation “200616857” which means “give little or no attention to”.

Encoding Semantic, Grammatical and Pragmatic Information

Finally, after representing all the previous forms of orthographically-represented information (bound and free morphemes), other subtle information are also extracted and are meticulously included in the understanding of words. This subtle information is in fact included along with the definitions of concepts in the natural language-UNL dictionary. They include semantic, grammatical and pragmatic features carefully selected from the tagset the UNL system employs. Semantic information such as abstractness, alienability, animacy, semantic typology (cognitive verb, communication verb, location, natural phenomena, etc.) and others are included with the entries representing each of the constituent concepts of a sentence. Figure 3 illustrates some of the semantic information assigned to the entry representing the concept “103906997” meaning “a writing implement with a point from which ink flows”; i.e., قلم حبر ‘pen’.

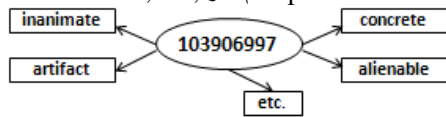


Figure 3. Some of the semantic information included with the concept for the Arabic lexical item قلم حبر in the Arabic-UNL dictionary

Moreover, entries in the natural language-UNL dictionary are also ascribed grammatical information. Grammatical information includes the concept’s lexical category, lexical structure, part of speech, transitivity, valency, case, syntactic role, etc. Figure 4 shows some of the grammatical features assigned to the conceptual representation “202317094” meaning “bestow, especially officially”; i.e. منح ‘grant’



Figure 4. Some of the grammatical information included with the entry for the Arabic word منح in the Arabic-UNL dictionary

In addition to the kinds of information acknowledged so far, there are also other pieces of information that are inseparable from any sort of understanding of a text. These types have to do with the pragmatic meaning of a sentence. It in-

cludes, for example, the situational context. The situational context would to the extent possible refer to every non-linguistic factor that affects the meaning of a phrase. The UNL, of course, cannot account for *all* of these factors but it can, however, detect and encode *some* of them. For example, an important element in the situational context is the role of a word in a sentence. Therefore, in the UNL framework, the main (starting) node of a semantic network is marked. Similarly, in passive constructions, the subject is indicated by a tag that specifies it as the *topic*.

A different sort of context markers are those indicating information on the external context of the utterance, i.e., non-verbal elements of communication, such as prosody, sentence and text structure, and speech acts. Linguistic context is also encoded; special tags denote the linguistic neighborhood of a word such as punctuation marks and anaphoric references.

Finally, a further type of extralinguistic information has to do with the social context of the sentence. When marked explicitly by the use of certain words or structures, information about the social context is also included in the understanding of a sentence showing, for example, social deixis (politeness, formality, intimacy, etc.) and register (archaism, dialect, slang, etc.) and others⁴. The acknowledgment and inclusion of all such tags is quite necessary to claim the ability to truly understand a natural language sentence. Besides, they must be tagged in order to support later retrieval (Dey, 2001) as will be shown in the section 3 of this paper.

2.2 Analyzing Sentences

Understanding and encoding the meanings conveyed by every single morpheme in a certain sentence is far from sufficient to constitute an understanding. A simple list of concepts and tags will be hardly comprehensible even for the native speaker. Grammar rules are required to link these morphemes into a semantic network that represents the meaning of the sentence as a whole.

Deducing the pure semantic meaning directly from a simple list of concepts can be deemed impractical if not impossible; hence, the UNL system has opted for the use of an intermediary stage that maps this list onto an elaborate syntactic tree structure. The ordering of constituents in this tree and the syntactic relations that link them

⁴ These tags can also be found at <http://www.unlweb.net/wiki/index.php/Attribute>

together can, subsequently, help point out the kind of semantic links the sentence implies.

After encoding the information carried by each lexical item in section 2.1, grammar rules use this information to carry out the process of determining the syntactic structure underlying the input sentence. The UNL grammar workbench is divided into two main categories: transformation grammar and disambiguation grammar. Transformation grammar comprises the rules capable of extracting the abstract meaning conveyed by the input sentence while disambiguation grammar involves lexically, syntactically and semantically disambiguating the natural language sentence in order to reach the most accurate UNL representation possible (Alansary et al., 2010). Both Transformation and disambiguation grammars involve several phases and types of rules⁵. Yet, this paper will not delve deeply into the details of these phases or types (they have been discussed before in Alansary et al., 2010; Alansary, 2010); this paper rather aims at demonstrating how these grammars are capable of handling and deciphering natural language phenomena.

Determining Syntactic Structure

A significant section of the UNL transformation grammar is devoted to transforming the incoming natural language list into an elaborate syntactic structure. To demonstrate this process, the Arabic sentence in (3) will be used as an example.

(3) منح الرئيس قلادة النيل لمجدي يعقوب

manaha ?arra?iisu qiladata ?anniili limagdii ya?quub 'The president granted the Nile Medal to Magdi Yacoub'

The information assigned in the previous stage will come into use here; transformation grammar rules use the grammatical, semantic and pragmatic information as guides to determine the syntactic position each morpheme holds in the syntactic structure of the sentence. For example, the rules in (4) transform the natural language sentence in (3) into the syntactic tree in figure 5.

(4a) (V, %01)(N,HUM, %02):=VS(%01;%02);

⁵ More information about this division and the phases involved is found in http://www.unlweb.net/wiki/index.php/Grammar_Specs

(4b) (V,%x)(N,NONHUM,%y):=VC(V,%x;N,%y);
(4c) (V,TST2,%01)PP("J";%02):=VC(%01;%02);

Rule (4a) specifies that when a verb is assigned the semantic feature "give verb" and is followed by a "human" noun, the noun is the syntactic specifier of the verb. Rule (4b), on the other hand, states that the syntactic relation between a "give verb" and a following "non-human" noun is a syntactic complementizer relation. Finally, a grammatical feature of the verb "منح"; it being "ditransitive", dictates that when being followed by a prepositional phrase headed by the preposition "J", the prepositional phrase is a second complementizer for that verb.

Along with these transformation rules, disambiguation rules are at work. Tree disambiguation rules also prevent wrong lexical choices and provoke best matches by determining which constituents can share a syntactic relation. For example, the rules in (5) help restrict the application of transformation rules by dictating that a prepositional complementizer (PC) following a ditransitive verb (TST2) can never be an adjunct for that verb (probability = 0) while it being a complementizer for that verb is very plausible (probability = 225) since a ditransitive verb inevitably requires two complements

(5a) VA(TST2,PC):=0;
(5b) VC(TST2,PC):=225;

The result of these processes would be the syntactic structure in figure 5.

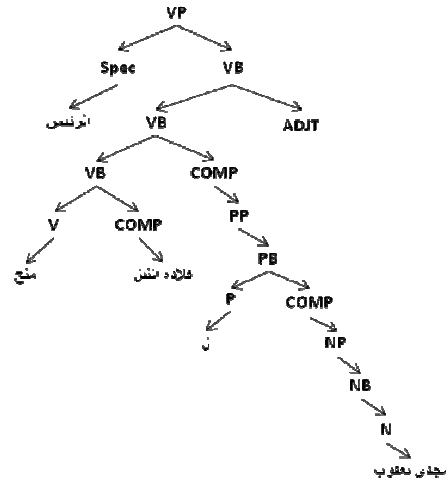


Figure 5. The deep syntactic structure of the Arabic sentence in (3)

Determining Semantic Structure

Finally, in order to generate the final understanding of an input sentence, different types of trans-

formation rules apply to transform the syntactic structure in figure 5 into a semantic network. This semantic network will incorporate all of the information extracted and defined in the previous stages. Using the same example sentence in (3), the functioning of the semantic analysis rules will be demonstrated. The rules in (6) use the syntactic relations as a guide to determine the semantic links between the concepts.

- (6a) $VS(\%01;\%02)=agt(VER,\%01;\%02,NOU);$
 (6b) $VC(\%01;\%02)=obj(VER,\%01;NOU,\%02);$
 (6c) $VC(\%01;PC("·\%";"ل")):=gol(VER,\%01;NOU,\%02);$

The rule in (6a) states that if the specifier of a verb is a noun syntactically, then the noun is the agent of the verb semantically while rule (6b) assigns a semantic object relation between two words that share a syntactic complementizer relation. In the same manner, rule (6c) states that if the complement of a verb is a noun syntactically and it is a prepositional phrase introduced by the preposition (ل), then the noun is the goal of the verb semantically.

Also on the semantic level, disambiguation rules (i.e. network disambiguation rules) apply over the network structure of UNL graphs to constrain the application of transformation rules. Disambiguation rules constrain the type of constituents to be a member in a binary semantic relation. However, in this example sentence, no network disambiguation rules were required.

All of the previous processes work in unison to finally generate the semantic network in figure 6.

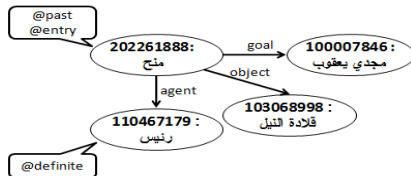


Figure 6. The semantic network representing the Arabic sentence in (3)⁶

3 Generating Language

A capable understanding framework is not only essential to the process of analyzing and processing natural language, Natural language Generation (NLG) can be deemed impossible if an adequate understanding foundation is unavail-

⁶ This semantic network only represents the main structure of the sentence in (3) and some of the most crucial tags; it does not incorporate all of the semantic, grammatical, and pragmatic information detected because of the space limitation.

able. An efficient NLG system must be able to generate language accurately in order to answer questions, for example, or interact with the user for the purposes of translation, information retrieval, etc.

In the following subsections, the UNL grammar workbench as an efficient and robust means for generating language will be considered. The process of generation may be seen to some extent as the mirror image of the analysis process; the abstract meaning stored inside the machine in the form of an extensive semantic network is transformed into a natural language sentence in two main stages. First, the whole structure of the sentence to be generated is determined on the deep level then on the surface level. Second, the word forms necessary to convey that meaning are generated to fill in the slots in this structure and the final changes are made to form a comprehensible well-formed natural language sentence.

Similar to the process of analyzing natural language, generating well-formed sentences has to pass through five stages of transformation rules, in addition to disambiguation rules; passing from the abstract semantic network to a syntactic representation from which the final natural language sentence is generated. This arrangement of phases is not the main focus here; it is rather to demonstrate the types of rules at work and how they are able to generate language from meaning efficiently as will be shown in the following subsections.

3.1 Generating Sentences

A syntactic structure is indispensable to constitute a well-formed target language structure. Thus, the UNL framework uses a set of formal rules to translate the pure semantic links that make up the abstract meaning representation (i.e., the UNL network) into syntactic relations.

Generating Syntactic Structure

There are two types of syntactic structure; the deep structure and the surface structure. The deep structure of a sentence represents its meaning but interpreted using syntactic tags rather than semantic ones. The surface structure, on the other hand, reflects the ordering of the constituents in the final natural language sentence.

In the process of forming a sentence's deep structure, grammar rules are devoted to mapping the semantic relations from the semantic network onto their equivalents in a syntactic tree. As an example, the semantic network in figure 6 requires the mapping rules in (7) to map the se-

semantic agent, object and goal relations onto their counterpart syntactic relations: verb specifier, verb complementizer and second verb complementizer, respectively.

- (7a) $\text{agt}(\text{VER}, \%01; \%02, \text{NOU}) := \text{VS}(\%01; \%02);$
 (7b) $\text{obj}(\text{VER}, \%01; \text{NOU}, \%02) := \text{VC}(\%01; \%02);$
 (7c) $\text{gol}(\text{VER}, \%01; \text{NOU}, \%02) := \text{VC}(\%01; \text{PC}(" \cdot \cdot \%"; " \text{ل}));$

The mapping rule in (7a) states that if the agent of a verb is a noun semantically, then the noun is the specifier of the verb syntactically and thus, occupies the specified positions in the syntactic tree. Similarly, the rule in (7b) maps the semantic object relation onto the position of a complementizer relation syntactically. Finally, rule (7c) maps the semantic goal relation onto the position of a (second) complementizer relation. The result, of course, would be the same syntactic structure in figure 5 above.

However, this deep structure does not always reflect the correct ordering of constituents in the final natural language sentence. The constituents of the tree have to be mapped onto a morphological sequence that is considered well-formed according to the grammar of the target language. This ordering is determined in the surface syntactic structure; thus, this deep syntactic structure has to be mapped onto a surface structure before being generated as a natural language sentence.

A sentence may have multiple surface structures since the same meaning may be reflected in several synonymous sentences. The Arabic language is especially abundant in such cases because Arabic word order is comparatively free; although the canonical order of an Arabic sentence is VSO (Verb-Subject-Object), most other orders can occur under appropriate circumstances (Ramsay and Mansour, 2006).

Thus, a different type of grammar rules is subsequently used to determine the exact position of a constituent with regards to the others, when certain conditions are fulfilled. For example, the rules (8) and (9) can generate two different equivalent versions of the syntactic structure in figure 5; these two versions are shown in figures 7 and 8, respectively.

- (8) $\text{VB}(\text{VB}(\%x; \%y); \%z) \text{VS}(\%x; \%v) := \text{VP}(\text{VB}(\text{VB}(\%x; \%v); \%y); \%z);$
 (9) $\text{VB}(\text{VB}(\%x; \%y); \%z) \text{VS}(\%x; \%v) := \text{VP}(\text{VB}(\text{VB}(\%x; \%v); \%z); \%y);$

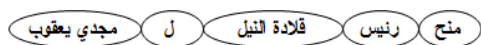


Figure 7. The first alternative surface structure for the deep structure in figure 5

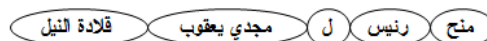


Figure 8. The second alternative surface structure for the deep structure in figure 5

The rule in (8) positions the second complement “مجدى يعقوب” before the first complement “قلادة النيل” to generate the sentence in figure 6. On the other hand, the rule in (9) positions them reversely to generate the sentence in figure 8.

Rules such as the previous apply generally to position constituents that behave regularly; nonetheless, in case of exceptions or categories that do not follow a regular distributional pattern the ordering of the constituent is informed in the dictionary. For example, the Arabic preposition ل ‘to’ is assigned the tag “IEMT” to indicate that, unlike most other prepositions, this preposition immediately precedes the noun following it, without and intervening blank space. This is indicated by the distribution rule in (10).

- (10) $\text{PB}(\text{PRE}) := \text{PB}(+\text{IBEF})$

Moreover, in special cases, the ordering specified in the surface structures needs to undergo some adjustment to reflect some aspect of meaning. In such cases, movement rules rearrange the constituents in a sentence to deal with transformations that affect only the surface structure of the sentence such as topicalization and passivization. For example, the movement rule in (11) changes active structures headed by monotransitive verbs into passive by changing the position of a the verb complementizer to fill the place of the verb specifier, while the verb specifier moves into the position of the verb adjunct as a prepositional phrases headed by the preposition بواسطة.

- (11) $\text{VC}(\% \text{head}; \% \text{comp}) \text{VS}(\% \text{head}; \% \text{spec}) := \text{VS}(\% \text{head}; \% \text{comp}) \text{VA}(\% \text{head}; \text{PC}([\text{بواسطة}]; \% \text{spec}));$

Generating Functional Morphemes

Up to this point in discussion, the natural language sentence is still an abstract list of concepts. As mentioned earlier, the semantic network is not only composed of nodes representing the main concepts and the semantic relations that tie these concepts together. Each node in this network is assigned numerous tags that signify the omitted closed-class free forms such as particles, prepositions, conjunctions, auxiliaries, interjections, quantifiers and others.

Closed-classes must also be acknowledged in the deep and surface structures of a sentence.

Therefore, parallel to the previous section, other types of rules are at work to express these closed-classes in the form of free morphemes, and position them in the syntactic structures being formed. For example, the rule in (12) generates and positions the Arabic definite article "ال" in a surface structure such as the one in figure 7 as illustrated in figure 9.

(12) @def := NS(DP([ال]));



Figure 9. The surface structure in figure 7 after generating and positioning the definite article "ال"

Moreover, in some cases and in some languages a grammatical feature has to be expressed independently as a free morpheme; a phenomenon called periphrasis. An example of this phenomenon is the present perfect tense in Arabic which is formed by adding the particle قد qad before the present verb. The rule in (13) generates this construction.

(13) VH(%vh,PRS,PFC):=+IC([قد];%vh,+PAS);

This rule states that the head of the verbal phrase receives the feature PTP (past simple) and becomes the complement of an inflectional phrase headed by the lemma قد if it has the features PRS and PFC (present and perfect).

In addition to grammatical and functional morphemes, some semantic relationships are expressed in some languages subtly through word order, inflection or derivation, while in other languages some relation has to be expressed in the form of free morphemes. Consequently, when generating Arabic sentences, some of the semantic relations used within the UNL framework had to be included in the syntactic structures as distinct Arabic lexical items. An example is the UNL semantic relation "material" which has to be expressed in Arabic as مصنوع من maSnuuEun min 'made of' to link between the object and the material. This is illustrated in the sentence قاميص قطن من القطن qamiiSun maSnuuEun min ?alquTni 'a cotton shirt' as shown in figure 10.

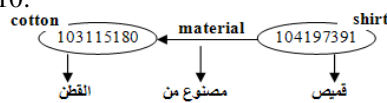


Figure 10. Generating the Arabic words مصنوع من to replace the semantic relation "material"

3.2 Generating Word Forms

In addition to the core concept conveyed via a natural language word, its superficial form in a sentence tells a lot about the role it plays in a sentence and the relationship it holds with other words. Moreover, incorrect word forms can deem a sentence incorrect and incomprehensible according to the grammatical regulations of the target language. Therefore, the final step to generating a well-formed natural language sentence is generating the required word form of each of the constituent lexical items. In this stage, grammatical information will be reflected on the form of the word itself, unlike the previous section where some grammatical or functional features had to be expressed as free morphemes.

Inflection

Inflection is the phenomenon of changing or modifying the form of a lexical item to indicate a certain grammatical feature it should convey. For example, verbs are inflected (or conjugated) to indicate their tense, aspect, etc. while nouns, adjectives and adverbs are inflected (or declined) to indicate their tense, mood, voice, etc. Inflection takes place mainly through the process of affixation; prefixation, infixation, suffixation or circumfixation.

The grammar workbench adopted in the UNL framework provides the means for generating the required morphological elements and attaching them to the intended concept. Inflection grammar is triggered by the tags put on the intended concept in the semantic network such as PLR (plural), FEM (feminine) or 2PS (second person).

Inflection is handled through inflectional paradigms in case of regular behavior, and inflectional rules in case of irregular behavior. These two may also work together in words that exhibit quasi-regular behavior. For example, a single inflectional paradigm (14) can handle the generation of the plural forms أواصر?awAsir 'relation' out of the singular أصرة?asirah 'relation' and the plural أواني?awaaniy 'pots' out of the singular أنية?aniyyah 'pot'.

(14) PLR:=[1]>"\,">";

This paradigm inserts the string "وا" after the first letter and deletes the final letter in the word to generate the plural forms.

On the other hand, an inflectional rule handles irregular inflections, and, is thus, only applicable to a single lexical item. An example is the Arabic word امرأة?imraah 'woman' of which the plural

is نساء nisaa' 'women'. This case is handled via the affixation rule in (15) which replaces the whole string امرأة with the string نساء.

(15) PLR:="نساء";

Arabic verbs are even more complex; a single verb may have over 60 distinct forms. However, the inflection of Arabic verbs is fairly regular and is, therefore, easily computed in the form of formal inflectional paradigms that can generate, for example, the forms of the Arabic verbs 'ask' and 'build' of which some are shown in (16) and (17), respectively.

(16) سأل- سألا - سئلا- يسألان- سئلا- سألوا- سئلا- يسألون-
سألت- سئلت- سألن- يسألن- سألت- سئلت- تسأل- سل- أنتما
سألتما

(17) بنى- يبني- بنيا - بينان- بنوا - بينون - بنت - تبني - بنتا -
تبنيان - بنين - بينين - بنيت - تبني - ابن

Agreement

Affixation rules assume the responsibility of generating all the required word forms to fit the tags explicitly marked in the semantic network. Nonetheless, in many cases, other constituents will imitate the tagged ones in some grammatical feature; a phenomenon that is called agreement. Agreement (or concord) is a form of cross-reference between different parts of a sentence or phrase, it happens when a word changes form depending on the other words it relates to. Special UNL grammar rules are devoted to determining which constituents receive or assign a grammatical feature and under what circumstances. (18) shows a simple conditional agreement rule. This rule specifies that an adjunct receives the gender from the noun if adjective.

(18) NA(ADJ):=NA(+RGEN);

A different kind of agreement of special importance to the Arabic language is that case marking. Usually a language is said to have inflectional case only if nouns change their form to reflect their case. Case marking is the process of assigning grammatical case values to dependent nouns for the type of relationship they bear to their heads.

The type of inflection an Arabic noun undergoes greatly depends on its case. A case-marking rule such as the one in (19) determines an adjective to be inflected for plural by adding the suffix "ين" rather than "ون" when modifying a noun in the accusative case.

(19)(%x,M500,MCL,ACC):=(%x,-
M500,+FLX(MCL&ACC:=0□"ين"););

Spelling changes

Other word-level changes in the form of a word may not depend on its structure or syntactic role in a sentence but rather on its linguistic neighborhood. Examples of such changes are changes in the spelling of a word as a result of contraction, assimilation, elision, etc. or capitalization in the beginning of a sentence (in Germanic languages) or the use of punctuation marks.

These kinds of transformations are handled by linear rules. Linear rules apply transformations over ordered sequences of isolated words in the UNL framework. Linear rules replace, add or delete certain characters in a word according to the contiguous characters from other words. For example, the Arabic definite article "ال" when preceded by the preposition 'ل'; the first letter from the definite article is deleted and the preposition immediately adheres to the remaining character from the definite article with no intervening blank spaces. The rule in (20) performs this process.

(20) (%x,M90,DFN,LAM):=(%x,-
M90,+FLX(DFN&LAM="ل">""););

4 Scope and Limitations

The UNL system currently supports the processing of 17 different languages. The main resources necessary for their analysis and generation (dictionaries and grammars) are being built by the respective institutions scattered all over the world. Yet, the UNL system is flexible enough to support any other natural language once the necessary resources are built.

These processing capabilities cover the morphological, semantic, syntactic and phonetic aspects of natural language texts. However, the phonetic module is not yet activated but will be in the near future. Also, the syntactic module is currently devoted to handling the basic syntactic structures; other more complex structures are to be focused on in later stages of development.

Nevertheless, the UNL workbench does not claim it represents the 'full' meaning of a word, sentence or text using these modules since 'full' meaning, as mentioned earlier, may depend on an infinite list of factors such as: intention, world knowledge, past experiences, etc. Although these factors are mostly known for the human speaker/writer and listener/reader, such factors are too subtle and subjective for any attempt of systematic processing.

Moreover, it must also be clear that the UNL system only represents the most 'consensual' meaning attributed to words and phrases, other

equivocal meanings are quite complex for a machine to infer. Thus, much of the subtleties of poetry, metaphors, and other indirect communicative behaviors are beyond the current scope of the system; the UNL system mainly aims at conveying the direct communicative meaning as it constitutes the most part of day-to-day communications.

Users can establish the validity of the UNL workbench by using it to process natural language phenomena. This has already been done by dozens of computational linguists in the various UNL language centers who are, at the present moment, using the workbench to produce the necessary resources. The workbench has been found sufficient, flexible and representative of the phenomena exhibited by the natural languages being handled.

5 Conclusion

A system capable of understanding natural language sentences is of potentially unlimited uses in the field of natural language processing. As this paper aimed to demonstrate, the UNL framework provides natural language processing experts with a vast array of tools and mechanisms that would aid them in the endeavor of reaching a true, full and accurate understanding of a natural language sentence. The most obvious application of this system is, of course, machine translation where the UNL semantic representation functions as an interlingua; however, machine translation is definitely not the only use. A language-neutral representation of meaning as opposed to syntactic matching should be of great use in areas such as cross-lingual information retrieval. Also, by distinguishing between main concepts and other secondary constituents, this system can be used in text summarization or text extension. Another fundamental use would be to use the understanding of texts as the source encoding for webpages which, upon request, can be generated in the natural language the user chooses.

References

- Alansary, Sameh, Magdy Nagi and Noha Adly. 2006. Generating Arabic Text: the Decoding Component in an Interlingual System for Man-Machine Communication in Natural Language. In *proceedings of the 6th International Conference on Language Engineering*, Cairo, Egypt.
- Alansary, Sameh, Magdy Nagi, and Noha Adly. 2006. Processing Arabic Text Content: The Encoding Component in an Interlingual System for Man-Machine Communication in Natural Language". In *proceedings of the 6th International Conference on Language Engineering*, Cairo, Egypt.
- Alansary, Sameh. 2010. A Practical Application of the UNL+3 Program on the Arabic Language. In *Proceedings of the 10th International Conference on Language Engineering*, Cairo, Egypt
- Bekios, Juan, Igor Boguslavsky, Jesús Cardeñosa and Carolina Gallardo. 2007. Using Wordnet for building an Interlingua Dictionary. In *proceedings of 5th International Conference on Information Research and Applications*, I.TECH. vol.1, pages 39-46, Varna, Bulgaria.
- Boguslavsky, Igor, Jesús Cardeñosa and Carolina Gallardo. 2008. A Novel Approach to Creating Disambiguated Multilingual Dictionaries. *Applied Linguistics*, vol. 30: 70-92.
- Boudhh, Sangharsh, and Pushpak Bhattacharyya. 2009. Unification of Universal Words Dictionaries using WordNet Ontology and Similarity Measures. In *proceedings of the 7th International Conference on Computer Science and Information Technologies*, CSIT 2009, Yerevan, Armenia.
- Dey, Anind K. 2001. Understanding and Using Context. *Personal and Ubiquitous Computing Journal*, vol. 5 (1): 4-7.
- Fellbaum, Christiane, editor. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.
- Francis, Wendy. S. 2005. Bilingual semantic and conceptual representation. In J. F. Kroll and A. M. B. de Groot, editors, *Handbook of bilingualism: Psycholinguistic approaches*. Oxford University Press, New York, NY, pages 251-267
- Liddy, Elizabeth D. 2001. Natural Language Processing. In *Encyclopedia of Library and Information Science*, 2nd Ed.: Marcel Decker, Inc., New York
- Martins, Ronaldo and Vahan Avetisyan. 2009. Generative and Enumerative Lexicons in the UNL Framework. In *proceedings of 7th International Conference on Computer Science and Information Technologies*, CSIT 2009, Yerevan, Armenia.
- Ramsay, Allan M. and Hanady Mansour. 2006. Local constraints on Arabic word order. In *Proceedings of 5th International Conference on NLP, FinTAL 200*), pages 447-457, Turku.
- Talmy, Leonard. 2000. *Toward a Cognitive Semantics*. Vol. I: *Concept structuring systems*. MIT Press, Cambridge

Evaluation of crowdsourcing transcriptions for African languages

Hadrien Gelas^{1,2}, Solomon Teferra Abate², Laurent Besacier², François Pellegrino¹

¹Laboratoire Dynamique Du Langage, CNRS - Université de Lyon, France

²Laboratoire Informatique de Grenoble, CNRS - Université Joseph Fourier Grenoble 1, France
{hadrien.gelas, francois.pellegrino}@univ-lyon2.fr, {solomon.abate, laurent.besacier}@imag.fr

Abstract

We evaluate the quality of speech transcriptions acquired by crowdsourcing to develop ASR acoustic models (AM) for under-resourced languages. We have developed AMs using reference (REF) transcriptions and transcriptions from crowdsourcing (TRK) for Swahili and Amharic. While the Amharic transcription was much slower than that of Swahili to complete, the speech recognition systems developed using REF and TRK transcriptions have almost similar (40.1 vs 39.6 for Amharic and 38.0 vs 38.5 for Swahili) word recognition error rate. Moreover, the character level disagreement rates between REF and TRK are only 3.3% and 6.1% for Amharic and Swahili, respectively. We conclude that it is possible to acquire quality transcriptions from the crowd for under-resourced languages using Amazon’s Mechanical Turk. Recognizing such a great potential of it, we recommend some legal and ethical issues to consider.

1 Foreword

This paper deals with the use of Amazon’s Mechanical Turk (MTurk) which is a subject of controversy among researchers for obvious legal and ethical issues. The goal of this paper is to evaluate the quality of the data produced via crowdsourcing and not to produce a mass of data for a low price (in this experiment, we have actually retranscribed speech data for which we already had transcriptions). Ethical issues on working with MTurk are discussed in the last section of this paper where guidelines of “good conduct” are proposed.

2 Introduction

Speech transcriptions are required for any research in speech recognition. However, the time and cost of manual speech transcription make difficult the collection of transcribed speech in all languages of the world.

Amazon’s Mechanical Turk (MTurk) is an online market place for work. It aims at outsourcing difficult or impossible tasks for computers called “*Human Intelligence Tasks*” (HITs) to willing human workers (“*turkers*”) around the Web. Taking use of this “crowd” brings two important benefits against traditional solutions (employees or contractors): repetitive, time consuming and/or costly tasks can be completed quickly for low payment.

Recently MTurk has been investigated as a great potential to reduce the cost of manual speech transcription. MTurk has been previously used by others to transcribe speech. For example, (Gruenstein et al., 2009; McGraw et al., 2009) report near-expert accuracy by using MTurk to correct the output of an automatic speech recognizer. (Marge et al., 2010b) combined multiple MTurk transcriptions to produce merged transcriptions that approached the accuracy of expert transcribers.

Most of the studies conducted on the use of MTurk for speech transcription take English as their subject of study which is one of the well resourced languages. The studies on English, including (Snow et al., 2008; McGraw et al., 2009), showed that MTurk can be used to cheaply create data for natural language processing applications. However, MTurk is not yet widely studied as a means to acquire useful data for under-resourced languages except a research conducted recently (Novotney and Callison-Burch, 2010) on Korean, Hindi and Tamil. On the other hand, there is a growing research interest towards speech and language processing for under-resourced and African languages. Specific workshops in this domain are appearing such as SLTU (Spoken

Languages Technologies for Under-resourced languages¹) and ALaT (African Language Technology²). Moreover, (Barnard et al., 2010a; Barnard et al., 2010b) highlighted interests using Automatic Speech Recognition for information access in Sub-Saharan Africa, with a focus on South-Africa.

In this paper we investigate the usability of MTurk for speech transcription to develop Automatic Speech Recognition (ASR) for two under-resourced African languages without combining transcription outputs. In Section 3, we review some of the works conducted on the use of MTurk for speech transcription. We then describe our experimental setups including the subject languages in Section 4. Section 5 presents the result of the experiment. Discussions and conclusions are presented in Section 6.

3 Related work

We find a lot of work on the use of MTurk in creating speech and language data (Marge et al., 2010b; Lane et al., 2010; Evanini et al., 2010; Callison-Burch and Dredze, 2010). It shows the increasing interests of the research community in the use of MTurk for various NLP domains such as collecting speech corpora as in (McGraw et al., 2010; Lane et al., 2010) and for speech transcription as in (Novotney and Callison-Burch, 2010; Evanini et al., 2010; Marge et al., 2010a)

Among the works, (Novotney and Callison-Burch, 2010) is the most related one to our study. The study investigated the effectiveness of MTurk transcription for training speech models and the quality of MTurk transcription is assessed by comparing the performance of one LVCSR system trained on Turker annotation and another trained on professional transcriptions of the same data set. The authors pointed out that average Turker disagreement to the LDC reference for Korean was 17% (computed at the character level giving Phone Error Rate-PER) and using these transcripts to train an LVCSR system instead of those provided by LDC decreased PER only by 0.8% from 51.3% to 52.1%. The system trained on the entire 27 hours of LDC Korean data obtained 41.2% PER.

Based on these findings, it is concluded that since performance degradation is so small, redundant annotation to improve quality does not worth

the cost. Resources are better spent collecting more transcription.

4 Experiment Description

4.1 Languages

Amharic is a member of the Ethio-Semitic languages, which belong to the Semitic branch of the Afroasiatic super family. It is related to Hebrew, Arabic, and Syrian. According to the 1998 census, it is spoken by over 17 million people as a first language and by over 5 million as a second language throughout different regions of Ethiopia. The language is also spoken in other countries such as Egypt, Israel and the United States. Amharic has its own writing system which is syllabary. It is possible to transcribe Amharic speech using either isolated phoneme symbols or concatenated CV (Consonant Vowel) syllabary symbols.

Swahili is a Bantu language often used as a vehicular language in a wide area of East Africa. It is not only the national language of Kenya and Tanzania but also spoken in different parts of Democratic Republic of Congo, Mozambique, Somalia, Uganda, Rwanda and Burundi. Most estimations give over 50 million speakers (with only less than 5 million native speakers). Structurally, Swahili is often considered as an agglutinative language (Marten, 2006). Even if non-total, it has typical Bantu features, such as noun class and agreement systems and complex verbal morphology. It was written with an Arabic-based orthography before it adopted the Roman script (standardized since 1930).

4.2 Corpora

Both Amharic and Swahili audio corpora were collected following the same protocol. Texts were first extracted from news websites and then segmented by sentence. Recordings were made by native speakers reading sentence by sentence with the possibility to re-record anytime they considered having mispronounced. The whole Amharic speech corpus (Abate et al., 2005) contains 20 hours of training speech collected from 100 speakers who read a total of 10850 sentences (28666 tokens). Still in its first steps of development, Swahili corpus corresponds to 3 hours and a half read by 5 speakers (3 male and 2 female). The sentences read by speakers were used as our gold standards to compare with the transcriptions obtained by MTurk. So the transcribed data were already available for control. We recall that the goal

¹www.mica.edu.vn/sltu-2010/

²aflat.org/

of this paper is to evaluate the quality of crowdsourcing tools to obtain good enough transcriptions for resource scarce languages.

4.3 Transcription Task

For our transcription task, we selected from the Swahili corpus all (1183 files) the audio files between 3 and 7 seconds (mean length 4.8 sec and total one hour and a half). The same number of files were selected from the Amharic corpus (mean length 5.9 sec). These files were published (a HIT for a file) on MTurk with a payment rate of USD 0.05 per HIT. To avoid inept Turkers, HIT descriptions and instructions were given in the respective languages (Amharic and Swahili). For the Amharic transcription to be in Unicode encoding, we have given the address of an online Unicode based Amharic virtual keyboard³ (Swahili transcriptions need no requirement).

5 Results

5.1 Analysis of the Turkers work

On such a small amount of sentences we chose to do the approval process manually via the MTurk web interface. Table 1 shows proportion of approved and rejected HITs for both languages. The higher rate of rejected HITs for Amharic can be explained by the much longer time the task was available for Turkers. We rejected HITs containing empty transcriptions, copy of instructions and descriptions from our HITs, non-sense text and HITs which were made by people who were trying to transcribe without any knowledge of the language. Doing this approval process manually can be considered as time consuming on a large amount of data. However, it was out of the scope of this paper to consider automated filtering/rejecting methods (this is part of future works). With the help of Mturk web interface directly allowing to reject or approve all works made by turkers known to do correct or incorrect work, this approval process took us only a few minutes each day (approximately 15min). Table 2 shows rejected HIT details.

Figure 1 shows the detailed completion rate per day for both languages. Among the 1183 sentences requested, Amharic has reached 54% of

³www.lexilogos.com/keyboard/amharic.htm

⁴This is the number of all the Turkers who submitted one or more Amharic HITs. It is not, therefore, the sum of the number of rejected and approved Turkers because there are Turkers who submitted some rejected HITs and some approved ones

	# workers	
	AMH	SWH
APP	12	3
REJ	171	31
TOT	177 ⁴	34

	# HITs	
	AMH	SWH
APP	589 (54.49%)	1183 (82.50%)
REJ	492 (45.51%)	250 (17.43%)
TOT	1081	1434

Table 1: Submitted HITs approval

Content of Rejected HITs	Percentage	
	Swahili	Amharic
Empty	92.86	60.57
Non-sense	3.17	20.33
Copy from instructions	1.98	5.70
Trying without knowledge	1.98	13.40

Table 2: Content of Rejected HITs

approved HITs in 73 days. On the other hand, Swahili was completed after 12 days showing a real variety of work rate among different languages.

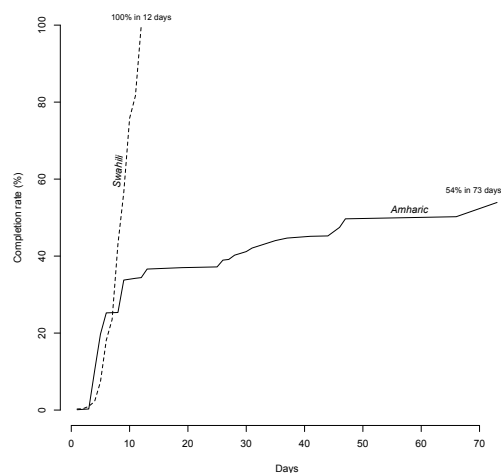


Figure 1: Completion rate per-day

One hypothesis for such a difference could simply be the effective population having access to MTurk. A recent survey (Ipeirotis, 2010) shows that 47% of the turkers were from the United States, 34% from India and the last 19% were divided among 66 non-detailed other countries. However, against this eventual demographic reason, we learn from U.S.ENGLISH⁵, that Swahili speakers are less numerous than Amharic speakers in the United States (36690 Swahili speakers against 82070 Amharic speakers).

⁵www.usefoundation.org/view/29

Moreover, Table 1 shows that numbers of workers doing coherent work was higher for Amharic than Swahili (12 and 3, respectively). Thus, a more likely reason would be the input burden for Amharic using the external virtual keyboard and copy/paste from another web page. The difficulty to do this while at the same time manage and listen to the audio file may have complicated the task and discouraged Turkers.

Nevertheless, HITs transcription productivity (Figure 2) indicates similar mean Turker productivities (15 and 17xRT for Amharic and Swahili, respectively). Obvious false values brought by some bias in *working time* indicated in MTurk results were removed (lower than 4xRT). Comparing with values in (Novotney and Callison-Burch, 2010), it is much less than historical high quality transcription rate (50xRT), but slightly more than MTurk transcriptions of English (estimated at 12xRT).

5.2 Evaluation of Turkers transcriptions quality

To evaluate Turkers transcriptions (TRK) quality, we computed accuracy against our reference transcriptions (REF). As both Amharic and Swahili are morphologically rich languages, we found relevant to calculate error rate at word-level (WER), syllable-level (SER) and character-level (CER). Besides, real usefulness of such transcriptions must be evaluated in an ASR system (detailed in 5.4). Indeed, some misspellings, differences of segmentation (which can be really frequent in morphologically rich languages) will not necessarily impact system performance but will still inflate WER (Novotney and Callison-Burch, 2010). The CER is less affected and, therefore, it reflects the transcription quality more than the WER. Our reference transcriptions are the sentences read during corpora recordings and they may also have some disagreements with the audio files due to reading errors and are imperfect.

Table 3 presents ER for each language depending on the computed level accuracy⁶. As expected, WER is pretty high (16.0% for Amharic and 27.7% for Swahili) while CER is low enough to approach disagreement among expert transcribers. The word level disagreement for a none agglutinative language ranges 2-4% WER (NIST, web).

⁶Five of the approved Amharic transcriptions and four of the Swahili ones were found to be not usable and were disregarded

The gap between WER and SER can be a good indication of the weight of different segmentation errors due to the rich morphology.

Amharic			
Level	# Snt	# Unit	ER
Word	584	4988	16.0
Syllable	584	21148	4.8
Character	584	42422	3.3
Swahili			
Level	# Snt	# Unit	ER
Word	1179	10998	27.7
Syllable	1179	31233	10.8
Character	1179	63171	6.1

Table 3: Error Rate (ER) of Turkers transcriptions

The low results for Swahili are clarified by giving per-Turker ER. Among the three Turkers who completed approved HITs, two have really similar disagreement with REF, 19.8% and 20.3% WER, 3.8% and 4.6% CER. The last Turker has a 28.5% WER and 6.3% CER but was the most productive and performed 90.2% of the HITs. By looking more closely to error analysis, it is possible to strongly suggest that this Turker is a second-language speaker with no difficulty to listen and transcribe but with some difference in writing to the reference transcription (see details in 5.3).

5.3 Error analysis

Table 4 shows most frequent confusion pairs for Swahili between REF transcriptions and TRK transcriptions. Most of the errors can be grouped into five categories that can also be found in Amharic.

Frq	REF	TKR
15	serikali	serekali
13	kuwa	kwa
12	rais	raisi
11	hao	hawa
11	maiti	maiiti
9	ndio	ndiyo
7	mkazi	mkasi
6	nini	kwanini
6	sababu	kwasababu
6	suala	swala
6	ufisadi	ofisadi
5	dhidi	didi
5	fainali	finali
5	jaji	jadgi

Table 4: Most frequent confusion pairs for Swahili.

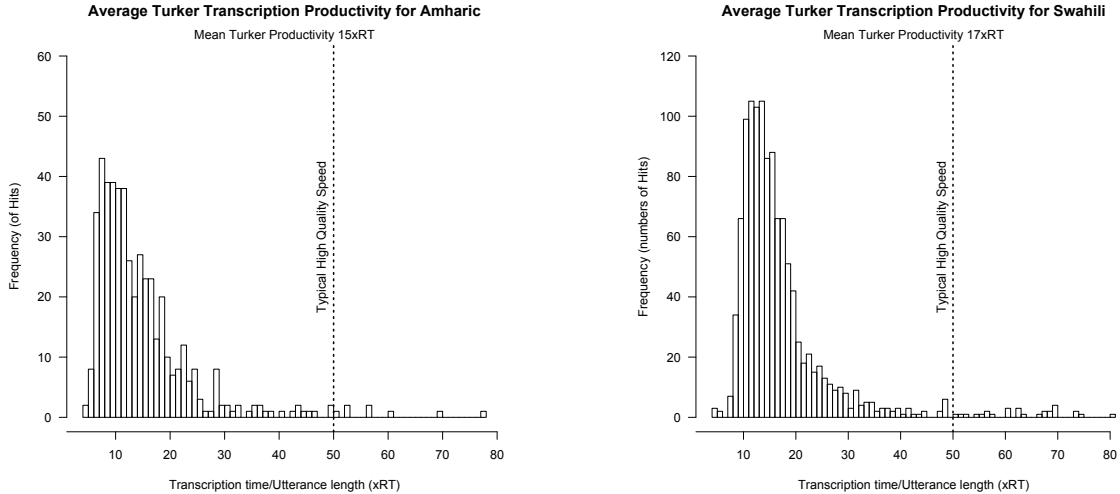


Figure 2: Histogram of HITs transcription productivity

- Wrong morphological segmentations: see words *nini*, *sababu*, both preceded by *kwa* in REF.
- Common spelling variations of words such as *serikali* and *rais* (sometimes even found in newspapers article); and misspellings due to English influence in loanwords like *fainali* and *jaji* (meaning final and judge).
- Misspellings based on pronunciation (see words *kuwa*, *ndio*, *suala*) and due to personal orthographic convention that can be seen in words *maiti*, *mkazi*, *ufisadi*, *dhidi*.

Errors in the last category were all made by the same Turker (the most productive one but having a high WER). Their frequency and regularity are the bases of our strong assumptions to consider this Turker as a second-language speaker. To illustrate this on the phoneme level, the phoneme [z] (voiced alveolar fricative always transcribed 'z' in Swahili) between vowels was always transcribed with an 's' as it is in other languages (like French or German). Similarly, phonemes [θ] and [ð] (dental fricatives transcribed 'th' and 'dh' in Swahili) were never recognized and may not be part of his consonant system.

5.4 Performance in Automatic Speech Recognition (ASR)

Considering the lack of data for Swahili, we used a very preliminary system. Based on a text corpus collected from 7 news websites (over 10 millions words), we built a statistical 3-gram language model using the SRI⁷ language model toolkit. Then, to generate a pronunciation dictionary, we

⁷www.speech.sri.com/projects/srilm/

extracted 64k more frequent words from the text corpus and automatically created pronunciations taking benefit of the regularity of the grapheme to phoneme conversion in Swahili. For Amharic, we have used the 65k vocabulary and the 3-gram language model that are developed by (Tachbelie et al., 2010).

We used SphinxTrain⁸ toolkit from Sphinx project for building Hidden Markov Models based acoustic models (AMs) for both languages. We trained context independent acoustic models of 36 and 40 phones for Swahili and Amharic, respectively. With the respective speech corpora used in the MTurk transcription task, we trained two (for each language) different AMs, one with REF transcriptions and the other using TRK transcriptions.

We computed WER using test sets which contain 82 and 359 utterances for Swahili and for Amharic, respectively. Table 5 presents the WER for both languages.

Languages	ASR	# Snt	# Wrđ	WER
Swahili	REF	82	1380	38.0
	TRK	82	1380	38.5
Amharic	REF	359	4097	40.1
	TRK	359	4097	39.6

Table 5: Performance of ASRs developed using REF and TRK transcriptions

Results indicate nearly similar performances for both languages with a slightly higher WER for the one based on TRK transcriptions (+0.5%) for Swahili and on the opposite direction for Amharic (-0.5%). This suggests, therefore, that non-expert transcriptions using crowdsourcing can be accu-

⁸cmusphinx.sourceforge.net/

rate enough for ASR. Moreover, not only for major languages such as English, languages from developing countries can also be considered. It also highlights the fact that even if most of the transcriptions are made by second-language speakers, it will not particularly affect ASR performances.

6 Discussion and Conclusions

In this study, we have investigated the usability of Amazon’s Mechanical Turk speech transcription for the development of acoustic models for two under-resourced African languages. The results of our study shows that we can acquire transcription of audio data with similar quality to a text that can be used to prepare a read speech corpus. However, all languages are not equal in completion rate. The two languages of this study clearly had a lower completion rate than English. And even among the languages of this study, Amharic’s task was not completed totally in a period of 73 days.

Thus, MTurk is proved to be a really interesting and efficient tool for NLP domains and some recommended practices were already proposed in (Callison-Burch and Dredze, 2010), mainly on how to be productive with MTurk. However, the use of this powerful tool also happens to be controversial among the research community for legal and ethical issues⁹. As in many fields of research, one should be careful on the manner the data are collected or the experiments are led to prevent any legal or ethical controversies. Indeed, it is often adopted that some charter or agreement need to be signed for any experiments or data collection; which is most of the time totally omitted by the requesters/turkers relationship in MTurk. In order to keep a research close to the highest ethical standards and attenuate these drawbacks, we propose a few guidelines of good conduct while using MTurk for research:

- Systematically explain “who we are”, “what we are doing” and “why” in HITs descriptions (as done traditionally for data collection);
- Make the data obtained available for free to the community;
- Set a reasonable payment so that the hourly rate is decent;
- Filter turkers by country of residence to avoid those who consider MTurk as their major source of funding.

⁹<http://workshops.elda.org/lislr2010/sites/lislr2010/IMG/pdf/W2-AddaMariani-Presentation.pdf>

References

- S. T. Abate, W. Menzel, and B. Tafila. 2005. An amharic speech corpus for large vocabulary continuous speech recognition. In *Interspeech*.
- E. Barnard, M. Davel, and G. van Huyssteen. 2010a. Speech technology for information access: a south african case study. In *AAAI Symposium on Artificial Intelligence*.
- E. Barnard, J. Schalkwyk, C. van Heerden, and P. Moreno. 2010b. Voice search for development. In *Interspeech*.
- C. Callison-Burch and M. Dredze. 2010. Creating speech and language data with amazon’s mechanical turk. In *NAACL HLT 2010 Workshop*.
- K. Evanini, D. Higgins, and K. Zechner. 2010. Using amazon mechanical turk for transcription of non-native speech. In *NAACL HLT 2010 Workshop*.
- A. Gruenstein, I. McGraw, and A. Sutherland. 2009. A self-transcribing speech corpus: collecting continuous speech with an online educational game. In *SLaTE Workshop*.
- P. Ipeirotis. 2010. Demographics of mechanical turk. *CeDER-10-01 working paper*. New York University.
- I. Lane, M. Eck, K. Rottmann, and A. Waibel. 2010. Tools for collecting speech corpora via mechanical-turk. In *NAACL HLT 2010 Workshop*.
- M. Marge, S. Banerjee, and A. Rudnicky. 2010a. Using the amazon mechanical turk for transcription of spoken language. In *ICASSP*.
- M. Marge, S. Banerjee, and A. Rudnicky. 2010b. Using the amazon mechanical turk to transcribe and annotate meeting speech for extractive summarization. In *NAACL HLT 2010 Workshop*.
- L. Marten. 2006. Swahili. In Keith Brown, editor, *The Encyclopedia of Languages and Linguistics, 2nd ed.*, volume 12, pages 304–308. Oxford: Elsevier.
- I. McGraw, A. Gruenstein, and A. Sutherland. 2009. A self-labeling speech corpus: Collecting spoken words with an online educational game. In *Interspeech*.
- I. McGraw, C. Lee, L. Hetherington, S. Seneff, and J. Glass. 2010. Collecting voices from the cloud. In *LREC’10*.
- The nist rich transcription evaluation project. <http://www.itl.nist.gov/iad/mig/tests/rt>.
- S. Novotney and C. Callison-Burch. 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *NAACL HLT 2010*.
- R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. 2008. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP’08*.
- M. Y. Tachbelie, S. T. Abate, and W. Menzel. 2010. Morpheme-based automatic speech recognition for a morphologically rich language - amharic. In *SLTU’10*.

Development of an Open Source Urdu Screen Reader for Visually Impaired People

Madiha Ijaz

National University of Computer and
Emerging Sciences, Lahore, Pakistan
madiha.ijaz@nu.edu.pk

Qaiser S. Durrani

National University of Computer and
Emerging Sciences, Lahore, Pakistan
qaiser.durrani@nu.edu.pk

Abstract

Speech technology has enabled computer accessibility for users with visual impairments but the language barrier poses a great challenge. This project is an effort to overcome the hurdles faced by visually impaired people, in terms of language barrier, by providing them access to digital information through software which can communicate with them in Urdu. A survey was conducted in schools for blind to assess their information and communication needs. The survey helped to deduce the learning abilities, competency level and usability requirements of visually impaired children. An open source screen reader, NVDA was localized and afterwards integrated with Urdu text-to-speech system. The system was deployed in a school of visually impaired children where they participated in training and testing of the system. Results showed that visually impaired children performed equally well and in some cases even better with the localized screen reader as compared to an English screen reader.

Keywords: Localization, assistive technology, visually impaired, screen reader, text-to-speech system, Non-Visual Desktop Access (NVDA)

1. Introduction

Technology has played a dramatic role in improving the lives of visually impaired people by providing them access to sources of information, i.e., newspapers, books etc. that were largely unavailable to them in past. The advancement in

information, communication and computer technology has resulted in development of specialized software known as screen readers, which simulate the human voice while reading the computer screen. English is used as communication language in most of the software. Work has started recently in localization area so that local blind community can benefit from the advancement in information and communication technology.

This project focuses on the development of an open source Urdu screen reader for visually impaired people of Pakistan and Urdu speaking community worldwide. An extensive field study was conducted on visually impaired people, specifically targeting school going children and their teachers; to assess their information and communication needs. On the basis of that survey, it was concluded that visually impaired community of Pakistan was in dire need of an Urdu screen reader that should be compatible with word processor so that visually impaired people could read and write Urdu text; secondly it should be compatible with a web browser so that visually impaired people could read web pages in Urdu and search desired information on the internet. To start with, the task of word processor compatibility was taken at first as it was more challenging and exciting. Once a visually impaired person is well versed in reading and writing Urdu text then web browser, email, chat client and other communication tools are easy to learn.

After conducting survey on screen readers in general and studying NVDA, JAWS, Window Eyes, HAL and Thunder Screen in detail; an open-

source screen reader Non-Visual Desktop Access (NVDA) was chosen. NVDA has already been localized into 20 languages. NVDA was localized in Urdu by following the guidelines available at NVDA website.

Urdu text-to-speech system has already been developed by Center for Research in Urdu Language Processing (CRULP). Urdu text-to-speech system was enhanced and improved according to the requirements of the screen reader and afterwards it was integrated with the localized screen reader NVDA.

The system was deployed in a school for visually impaired children for final testing, training and evaluation. Training and evaluation was carried out by defining competency level for word processor i.e. Microsoft Word for visually impaired children of 9th and 10th grade. A test was conducted before training and on the basis of results; strengths and weaknesses of the individuals were identified. Training was planned accordingly and competency level of children was established. Another test was conducted after training and hence the competency level of the children was assessed again after training.

2. Information and communication need assessment

According to Sight Savers International (Sight Savers, 2011), there are 1.4 million visually impaired people in Pakistan. No research has been conducted in the past to assess the ICT needs of visually impaired community. They are not taught computers in schools, except few where measures are being taken to make them computer literate.

The survey was conducted, primarily to understand the psychology of the visually impaired people and identify their information requirements, as these tasks are important for designing the system, training material and training methodology. The research was intended to assess the following areas regarding visually impaired

- Learning capabilities of visually impaired as compared to normal students
- Learning capabilities in computer usage
- Attitude towards learning aids
- Average age, when he/she can start learning computer.

- Comprehension through listening
- Preference regarding the source of information

The survey was conducted in small number of schools, both from public and private sector, for blind children. The target group included visually impaired children from grade 5th to 10th. They were divided in two groups; one group comprised of children who were not taught computer in school and hence they had either no knowledge or very little know how of computers. The other group comprised of children who were taught computer as a subject in the school; so they had basic computer skills and were familiar with the concept of screen reader, text-to-speech system, magnifiers etc.

Apart from conducting the survey on the visually impaired children; we also contacted their teachers, parents and organizations working for visually impaired people, e.g. Special Education Department, Pakistan Association for Blind, Baseerat Foundation, etc. to better understand the behavioral pattern and ICT needs of visually impaired people.

2.1. Observations

Various observations were made during visits to schools of visually impaired children. Summary is given below

- They are highly motivated.
- Their attitude towards life is very positive.
- They are eager to learn and those students who were not taught computer were desperately looking forward to have that knowledge.
- They share their requirements openly e.g. students who were using screen reader JAWS, complained about its verbosity etc.
- They can easily use mobile phones. They can easily dial a number, write an SMS and send to their friends. They also get updates regarding cricket matches from their mobile phones.
- The attitude of society is overall negative towards them and it results in low confidence level among visually impaired children.
- Hearing is used as the primary source for information gathering in case of visually impaired people. The other extensively used source of data is the sense of touch. Also, it has been observed that through hearing, a properly trained visually impaired person may

understand what is being taught to him/her much more quickly than a normal person.

- The most obvious weakness that is inherent in the visually impaired is the lack of visual input. Most of the interfaces that are encountered in daily life incorporate visual cues. These visual cues are useless for the visually impaired people as these people cannot function normally unless there are some other cues such as audio or something based on the sense of touch is provided.

3. Urdu Screen reader

Over the past few years, the amount of data available digitally has grown rapidly. Internet has become the primary source of information. Unfortunately, the visually impaired community cannot access the information available digitally. There is a limited number of software available to help visually impaired people access the digitally available data; in addition, the language of communication in these software is English. So, the visually impaired people of Pakistan, who do not know English, cannot access the digital data. To provide them this access, a software is required which can communicate with them in Urdu or other local languages.

3.1. Localization of screen reader

Various screen readers were analyzed in this project e.g. NVDA (2009), JAWS (2009), Window Eyes (2009), SuperNova formerly HAL (2009) and Thunder Screen (2009); in order to observe their performance in Microsoft Word, Microsoft Excel and Internet Explorer. Apart from these applications, these screen readers were also observed for how they respond to basic operations performed on computer.

Among these screen readers Non-Visual Desktop Access (NVDA) was chosen for the following reasons

- It is an open-source screen reader.
- It has been localized in 20 languages, which include Brazilian Portuguese, Croatian, Czech, Finnish, French, Galician, German, Hungarian, Italian, Japanese, Portuguese, Russian, Slovak, Spanish, Traditional Chinese, Afrikaans,

Polish, Thai, Ukrainian and Vietnamese (NVDA User guide, 2009).

- It is compatible with Microsoft Windows and is compatible with following applications
 - Mozilla Firefox.
 - Mozilla Thunderbird.
 - Early support for Microsoft Internet Explorer
 - Basic support for Microsoft Outlook Express / Windows Mail
 - Basic support for Microsoft Word and Excel
 - Support for accessible Java applications
 - Early support for Adobe Reader
 - Support for Windows Command Prompt and console applications

NVDA was localized in Urdu by following the guidelines available at NVDA website (Translating NVDA, 2009). The interface of NVDA was translated using Poedit (Poedit, 2009). A snapshot of localized NVDA is shown below in Figure 1.

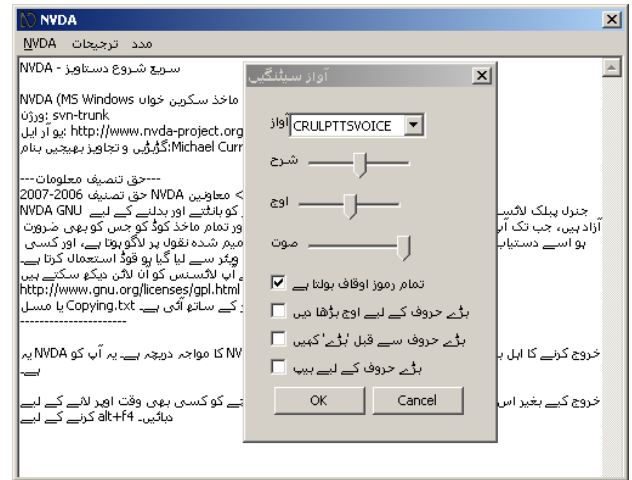


Figure 1: Localized NVDA

3.2. Urdu Text-to-speech system

Urdu text-to-speech system has already been developed by Center for Research in Urdu Language Processing (see Hussain (2004) and Hussain (2005) for details). Its features include

- Natural Language Processor
- Urdu speech synthesizer
- Basic duration model
- Perceptually tested diphone database of over 5000 diphones.

- Basic diacritics prediction
- Web page and email reader applications
- Lexicon of 80,000 words

One of the challenges faced during integration of screen reader and text-to-speech system was that text-to-speech systems are not multilingual; hence Urdu TTS discarded English text which is an integral part of interfaces nowadays. So we developed English to Urdu transliteration system and incorporated it with the Urdu TTS. The architecture of the transliteration system is shown below in Figure 2 (see Ali and Ijaz, 2009 for details).

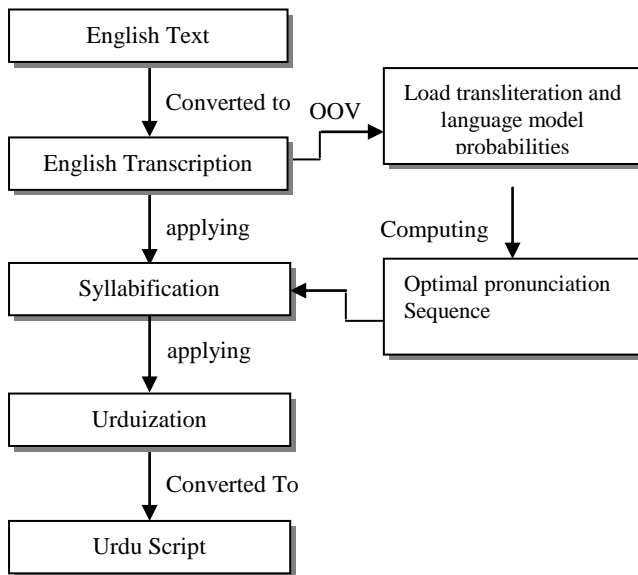


Figure 2: Architecture of English to Urdu transliteration system

Second major challenge was absence of rate control as one could not speed up or slow down the rate of speech. So rate variation control was incorporated and now listener could adjust rate of speech according to his/her preference.

3.3. Integration of text-to-speech with localized screen reader

NVDA has been translated in over 20 languages in which Urdu is not included. It can support any language by integrating that language speech synthesizer with it and by translating the interface and commands. Urdu text-to-speech system, which is an Urdu speech synthesizer application, was

integrated with NVDA through Microsoft Speech API (SAPI). SAPI provides standard interfaces for speech synthesis to communicate with TTS. NVDA provides interfaces to control its functionality through SAPI like volume, rate, pitch etc. SAPI query its method to determine which real-time actions to perform; Urdu TTS call this method frequently during rendering process to be as responsive as possible. SAPI method creates two threads to its audio reader and invokes Urdu TTS. First thread read Unicode based Urdu text and buffer it in audio text, the other thread read that speech data and pass it to SAPI. The architecture of the system is shown below in Figure 3.

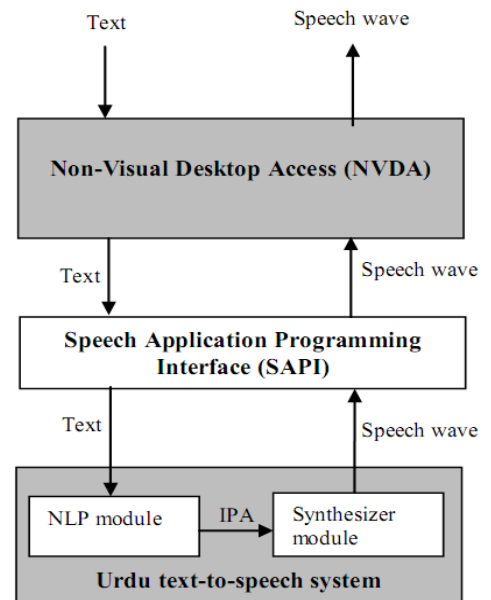


Figure 3: Architecture of the system

4. Training and evaluation

Visually impaired children of 9th and 10th grade of a local school with basic knowledge of computer and word processing, as it was part of their curriculum, were chosen for training. Students were interviewed and data summary of the students is described below in Table 1.

Total no of students	12 (5 completely blind and 7 partially sighted)
Age	12-16 years
Class	9th & 10th grade
Source of information	Braille

Hands on computer experience	3-4 years on average (1-2 hrs per week)
Assistive Technology	Screen reader (JAWS)
Input device	keyboard
Applications used	MS Word

Table 1: Data summary of visually impaired students

4.1. Competency level

For evaluation purposes, competency level was defined after consulting teachers who had taught visually impaired children of this age group. These levels are summarized below:

Level 1: Student is aware of the concept of word processing, knows how to open and exit MS Word

Level 2: Student can type text, traverse in the document and navigate menus

Level 3: Student can edit text

Level 4: Student is capable of opening, closing and saving document from/to desired path

Level 5: Student can format text

Level 6: Student can spell check document

Level 7: Student can find/replace text in document

A pre-training and post-training test was designed on basis of competency level. The pre-training test was taken on MS Word (English interface) with help of JAWS. Afterwards students were provided training on MS Word (Urdu Interface) using NVDA. A post-test was conducted after training and results of both tests were compared to see if there was any anomaly in the use of Urdu screen reader.

4.2. Pre-training test

A pre-training test was conducted in order to assess the competency level of the students with JAWS. They were asked to perform operations in MS Word (interface in English), e.g. type text, cut, copy, paste, spell check, save document and find/replace. Results are shown in Table 2. Total score was 16 and time was recorded (in minutes) for typing a 75 word English paragraph.

4.3. Training

Training was conducted for 5 days (30 minute session daily) in which 12 students participated. Students were trained on Urdu screen reader, i.e., NVDA and MS Word localized in Urdu. They

were taught Urdu keyboard and were asked to type Urdu text in MS Word. Most of the key mappings were same for English to Urdu Braille so according to them it was easy to memorize the Urdu keyboard. Afterwards, they were taught how to edit text; open, close, save, find/replace dialog box and spell checker.

4.4. Post-training test

A post-training test was conducted in order to assess the competency level of the students with NVDA. They were asked to perform operations in MS Word (interface in Urdu) e.g. type text, cut, copy, paste, spell check, save document and find/replace to see if they can perform them with equal proficiency as with JAWS and MS Word (interface in English). Results have been shown in Table 2. Total score was 16 and time was recorded (in minutes) for typing a 75 word Urdu paragraph.

5. Results

The pre-test and post-test results have been summarized in Table 2.

	Pre-test		Post-test	
	Overall Score	Time taken	Score	Time
Student #1	14	9	14	9
Student #2	10	18	12	20
Student #3	14	7	14	8
Student #4	14	12	14	9
Student #5	12	10	14	8
Student #6	10	20	11	22
Student #7	8	10	10	8
Student #8	12	10	14	8
Student #9	13	13	14	12
Student #10	12	10	14	7
Student #11	10	16	11	18
Student #12	12	12	13	10

Table 2: Pre-training and post-training test results

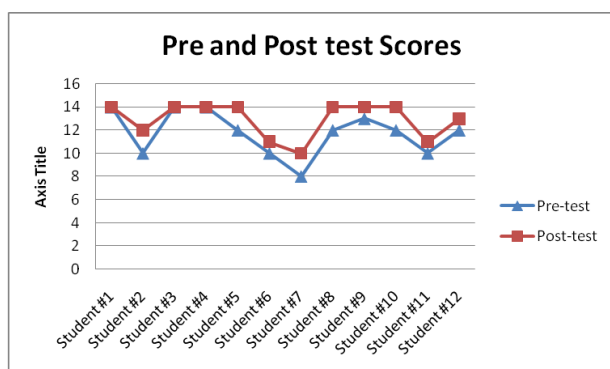


Figure 4: Pre and post-training test scores compared

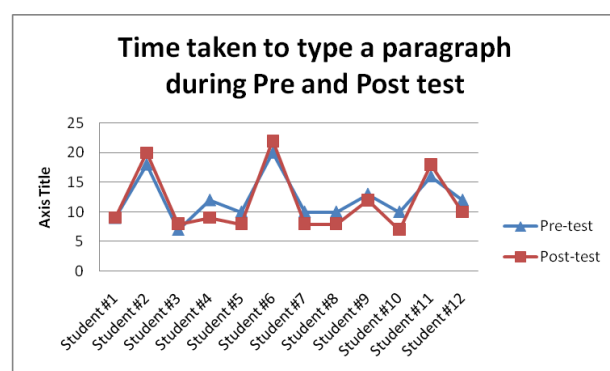


Figure 5: Pre and post-training test time compared for typing a 75 word paragraph

6. Conclusion

As it is shown in Table 2 and Figure 4, the post-training test scores of all the students are either same or have been improved as compared to the pre-training test, which shows that students were able to perform equally well with Urdu screen reader i.e. NVDA and MS Word localized in Urdu. Similarly, the time taken to type an English paragraph in pre-test and the time taken to type an Urdu paragraph in post-test show similar trend except for the few individuals whose typing skills were not good as shown in Figure 5.

Overall, the students were motivated and quite excited about learning how to read/write Urdu text. They even asked us to switch to localized Window XP i.e. Window XP with Urdu support. They were comfortable with the voice of the Urdu text-to-speech system although they sometimes complained regarding the verbose Urdu words used in interface of Microsoft Word.

7. Future Work

We intend to enhance the screen reader for web browsing, email and chat client. Afterwards we intend to develop training material and training methodology for visually impaired children in order to train them on these tools and then evaluate them in order to investigate the sustainability and scalability of this model.

Acknowledgement

This project has been funded by National University of Computer and Emerging Sciences, Pakistan.

The authors are also thankful to the administration, staff and students of Aziz Jehan Begum Trust school for blind for allowing them to conduct training and evaluation in that institute.

References

- Ali, Abbas Raza and Ijaz, Madiha. 2009. *Urdu to English Transliteration System*. Proceedings of the Conference on Language and Technology (CLT09), Lahore, Pakistan.
- Hussain, Sarmad. 2004. *Letter-to-Sound Rules for Urdu Text to Speech System*. Proceedings of Workshop on Computational Approaches to Arabic Script-based Language, COLING-2004, Geneva, Switzerland.
- Hussain, Sarmad. 2005. *Phonological Processing for Urdu Text to Speech System*. Yadava, Y, Bhattarai, G, Lohani, RR, Prasain, B and Parajuli, K (eds.) Contemporary issues in Nepalese linguistics. Katmandu, Linguistic Society of Nepal.
- Job Access with Speech (JAWS): Screen reader. Last accessed September 2009. <http://www.freedomscientific.com/products/fs/jaws-product-page.asp>
- Non-Visual Desktop Access (NVDA): Screen reader, 2008-2011. Last accessed September 2009. www.nvda-project.org
- Non-Visual Desktop Access (NVDA) User guide: 2008-2011. Last accessed September 2009.

www.nvda-project.org/documentation/nvda_0.6p2_userGuide.html

Poedit: Last accessed August 2009.
<http://www.poedit.net>

Sight Savers International: Last accessed January 2011. www.sightsavers.org

SuperNova: Screen reader. Last accessed September 2009. www.yourdolphin.com

Translating Non-Visual Desktop Access (NVDA): 2008-2011. Last accessed December 2010.
www.nvda-project.org/wiki/TranslatingNVDA

Thunder: Screen reader. Last accessed September 2009. www.screenreader.net

Urdu Text-to-speech system: 2004-2007. Last accessed December 2010.
www.crupl.org/software/langproc/TTS.htm

Window Eyes: Screen reader, 1990-2010. Last accessed September 2009. www.gwmicro.com

Continuous Sinhala Speech Recognizer

Thilini Nadungodage

Language Technology Research Laboratory,
University of Colombo
School of Computing, Sri Lanka.
hnd@ucsc.lk

Ruvan Weerasinghe

Language Technology Research Laboratory,
University of Colombo
School of Computing, Sri Lanka.
arw@ucsc.lk

Abstract

Automatic Speech Recognition has been successfully developed for many Western languages including English. Continuous, speaker independent speech recognition however has still not achieved high levels of accuracy owing to the variations in pronunciation between members of even a single community. This paper describes an effort to implement a speaker dependent continuous speech recognizer for a less resourced non-Latin language, namely, Sinhala. Using readily available open source tools, it shows that fairly accurate speaker dependent ASR systems for continuous speech can be built for newly digitized languages. The paper is expected to serve as a starting point for those interested in initiating projects in speech recognition for such 'new' languages from non-Latin linguistic traditions.

1 Introduction

Speech recognition has been a very active research area over the past few decades. Today, research on Speech Recognition has matured to a level where Automatic Speech Recognition (ASR) can be successfully implemented for many languages. Speech recognition systems are increasingly becoming popular because these systems create a friendly environment to the computer user. Speech recognition systems are able to provide an alternate and natural input method for computer use, particularly for the visually impaired. It could also potentially help to increase the overall productivity of general users by facilitating access programs and information more naturally and effectively.

In simple terms, speech recognition is the process of converting spoken words to machine-readable input. In more technical terms this can be stated as the process of converting an acoustic signal, captured by a microphone or a telephone, to a stream of words (Cole et al, 1996; Kemble, 2001).

Based on the two main types of human speech, speech recognition systems are generally classified into two types: discrete and continuous. In discrete speech, the spoken words are isolated. This means that in discrete speech, the speaker utters words in a way, which leaves a significant pause between words. Discrete speech recognition systems are created to recognize these isolated words, combination of words, or phrases and are referred to as Isolated (word) Speech Recognition (ISR) systems.

In continuous speech, the speaker pronounces words, phrases or sentences in a natural flow, so that successive words are dependent on each other as if they are linked together. There are no pauses or gaps between the spoken words in continuous speech. Continuous Speech Recognition (CSR) systems have developed to identify naturally flowing speech. The operation of a CSR system is more complex than an ISR system because they have to model dependencies between words.

Most of the speech recognition systems that have been developed so far are for English speech recognition. There is, however a lack of research in the field of recognizing non-Latin speech including many Indic languages and Sinhala. Sinhala is the mother tongue of majority of the Sri Lankans. It belongs to the Indo-Aryan branch of the Indo-European languages. Sinhala is also one of the official and national languages of Sri Lanka. Since there are many people in Sri Lanka who use Sinhala to communicate, there is

a need to pay attention to the research area of recognizing Sinhala speech. When considering the existing domain of Sinhala speech recognition, almost all the researches that have been done so far are on discrete speech recognition. This is due to the difficulties of separating words in continuous speech and collecting sufficient sample data.

This paper presents the results of a research work carried out to recognize continuous Sinhala speech. The objective of this research was to apply existing continuous speech recognition mechanisms to develop a continuous Sinhala speech recognizer, which is not bound to any specific domain.

The rest of the paper is organized as follows. Section 2 overviews the works related to the speech recognition domain. Section 3 gives the design of the ASR system. Section 4 relates the implementation of the ASR. Section 5 presents the evaluation of the recognizer using error rates and live inputs. Finally Section 6 draws overall conclusion and describes possible future work.

2 Related Work

Interest in ASR steadily progressed from 1930s when a system model for speech analysis and synthesis was proposed by Homer Dudley of Bell Laboratories (Dudley et al, 1939). This system model was called the Vocoder. The intention of the originally developed Vocoder was to act as a speech coder for applications in the area of telecommunication, at that time. Vocoder was mainly involved in securing radio communication, where the voice has to be encrypted before transmission. As the time passed with the evolution of the technology, the Vocoder has also further developed and modern Vocoder are used in developing many applications for areas like linguistics, computational neuroscience, psychophysics and cochlear implant.

After Homer Dudley's Vocoder, several other efforts have been carried out in the area of designing systems for ASR. The early attempts of such research were mainly conducted based on the theory of acoustic-phonetics (Juang and Rabinar, 2005). Most of the early speech recognition researches were conducted by concentrating on recognizing discrete speech. In 1952, three researches at Bell Laboratories, Davis, Biddulph and Balashek built a speaker dependent system to recognize digits which were uttered as isolated words (Davis et al, 1952). Another discrete speech recognizer was developed by Olson and

Belar of RCA Laboratories. This system was a speaker dependent system and was capable of recognizing 10 syllables (Olson and Belar, 1956). In 1959 J.W. Forgie and C.D. Forgie at MIT Lincoln Lab built a speaker independent recognizer for recognize ten vowels (Forgie and Forgie, 1959).

In 1960s some Japanese laboratories proposed designs to build special hardware for the task of speech recognition. Among these the phoneme recognizer built by Sakai and Doshita at Kyoto University was the first to employ the use of a speech segmenter. This includes segmenting the input speech wave into several portions and analyzing each portion separately (Sakai and Doshita, 1962).

The idea of the Hidden Markov Model (HMM) first came out in late 1960s (Rabiner and Juang, 1986; Juang and Rabiner, 1991). An HMM was referred to as a probabilistic function set of a Markov chain. In 1980s at the Bell Laboratories, the theory of HMM was used to improve the recognition accuracy of recognizers which were used particularly for speaker independent, large vocabulary speech recognition tasks.

The concept of Artificial Neural Network (ANN) was reintroduced in late 1980s. A neural network is a software model which simulates the function of the human brain in pattern recognition. Early attempts of using ANNs for speech recognition were based on simple tasks such as recognizing a few words or phonemes. Although ANNs showed successful results with these simple tasks, in their original form they were found to be not suitable for handling complex speech recognition tasks.

In most speech recognition research up to 1980s, converting a speech waveform into separate words, which is the first step of the process of recognizer understanding human speech, was considered as a major problem.

As Juang and Rabinar (2005) shows, the researchers learned two important facts when the speech recognition field evolved. The first fact is that although the speech recognizers were developed using grammatical constraints in a language, the users mostly speak natural sentences which have no grammar and the inputs to these systems are often corrupted by various noise components. As response to this factor a keyword spotting method was introduced.

The second is that, as in human-to-human speech communications, speech applications often required a dialog between the user and the

machine to reach some desired state of understanding. Such a dialog often required such operations as query and confirmation, thus providing some allowance for speech recognition and understanding errors.

In late 1990s, real speech enabled applications were finally developed. Microsoft Windows XP and Windows Vista developed speech recognition systems for personal computers used in daily life (Microsoft). Also these applications were available not only for English language but also for many other languages. Although these ASR systems do not perform perfectly, they are already delivering real value to some customers.

3 Design of ASR

Building of an ASR system mainly consists of designing two models, namely the Acoustic Model and the Language Model. The Acoustic model is responsible for detecting phonemes which was spoken and the Language Model is responsible for detecting connections between words in a sentence. The following sections give the design of these two models.

3.1 Acoustic Model

An acoustic model is created using audio recordings of speech and their text scripts and compiling them into a statistical representation of sounds which make up words. This is done through modeling the HMMs. The process of acoustic modeling is shown in Figure 1.

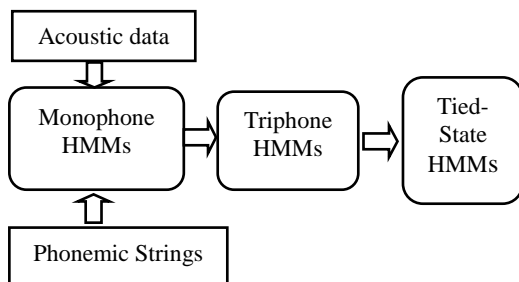


Figure 1. Block diagram of the acoustic modeling process.

3.2 Language Model

The way the words are connected to form sentences is modeled by the language model with the use of a pronunciation dictionary. The Language model of the proposed system is a statistical based language model as described in Rosenfeld (2000).

By assuming that the next word in the sequence will depend only upon one previous

word, a bigram (2-gram) language model is created. Finally using this bigram language model a network which contains words in the training data is created. The process of language modeling is shown in Figure 2.

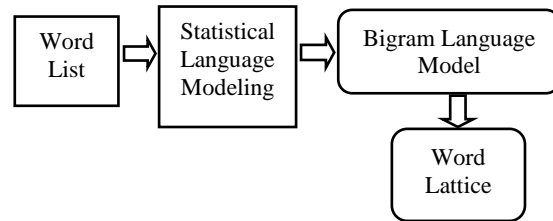


Figure 2. Block Diagram of the Language Modeling process.

3.3 Training

The basic procedure of building an ASR can be described as follows: the acoustic data of the training set goes through a feature extraction process and these features will be the input to the acoustic model training. The text transcription of the training data set is the input to build the language model. Trained acoustic model along with the language model is said to be the trained ASR system. The process of training the ASR is described in Figure 3. Next, the trained model goes through a testing process and the results obtained are used to adjust the trained model in order to get better and more accurate results.

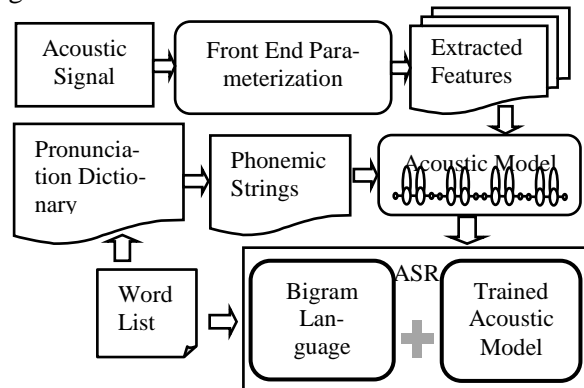


Figure 3. Block diagram of the process of training the ASR.

4 Implementation

This section describes the implementation of the ASR using Hidden Markov Model Toolkit (HTK) which was developed by the Cambridge University, UK (Young et al, 2006). HTK is primarily designed for speech recognition using Hidden Markov Models (HMMs). The steps of constructing the ASR can be divided a follows:

4.1 Data Collection

Unlike the English language, in Sinhala, written Sinhala and spoken Sinhala differ in some ways. While the grammar of written Sinhala depends on number, gender, person and tense, the grammar of spoken Sinhala does not follow them. Spoken Sinhala may vary in different geographical areas. So if we get the whole area of Sinhala language including both spoken and written Sinhala, we have to cover a huge vocabulary which is a very difficult and time consuming task. Hence, instead of covering the entire Sinhala vocabulary this research aims only the written Sinhala vocabulary, which can be used in automatic Sinhala dictation systems.

Before developing the ASR system, a speech corpus should be created and it includes recording continuous Sinhala speech samples. To build a better speech corpus the data should be recorded from various kinds of human voices. Age, gender, dialects and education should be the parameters which have to consider on collecting various voices. This requires a huge amount of effort and time. However this is the first attempt on building a continuous speech recognizer for Sinhala language and therefore as the initial step the data recording was done with a single female voice.

The first step in data collecting process is to prepare the prompt sheets. A prompt sheet is a list of all the sentences which need to be recorded. This sheet should be phonetically rich and cover almost all the phonetic transitions as possible. The prompt sheet was created using newspaper articles and the help of the UCSC 10M Sinhala Corpus. The prepared prompt sheet contained 983 distinct continuous Sinhala sentences for training purpose and these sentences were based on the most frequent words in Sinhala. The size of the vocabulary of data is 4055 words. For testing and evaluation purpose another 106 sentences were generated using the words contained in the previous set.

The prepared set of sentences was recorded using the software called *praat* with the sample frequency of 16 kHz using a Mono channel. Each of the training utterances was recorded three times. The recorded files were saved in the *.wav format. The recording process was carried out in a quiet environment but they were not 100% without surrounding noise. This problem can be treated as negligible since both training and testing data are recorded in the same envi-

ronment and therefore the noise will affect both data sets in an equal manner.

4.2 Data Preparation

The next step was to create the pronunciation dictionary. All the words used for the recordings are listed along with their phonetic representations in the pronunciation dictionary. Weerasinghe et al (2005) describes the phonemic inventory of the Sinhala language.

To train a set of HMMs every file of training data should have an associated phone level transcription. To make this task easier, it is better to create a word level transcription before creating the phone level transcription. The word level transcription was created by executing the Perl script *prompts2mlf* provided with the HTK toolkit using the previously prepared prompt sheet as input.

Using the created word level *MLF*, phone level transcription was created using the HTK label editor, *HLEd*. *HLEd* works by reading in a list of editing commands from an edit script file and then makes an edited copy of one or more label files. A *HLEd* edit script was used to insert the silence model 'sil' at the beginning and the end of each utterance.

Next, the features to be used in the recognition process should be extracted from the recorded speech signals. Feature extraction was performed using the *HCopy* tool. Here Mel Frequency Cepstral Coefficients (MFCC_D_A_E - 12 mel-cepstral coefficients, 12 delta coefficients, 12 acceleration coefficients, log energy, delta energy, and acceleration energy) was used to parameterize the speech signals into feature vectors with 39 features.

Building an ASR consists of creating two major models, Language model and the Acoustic model. The way the words are connected to form sentences is modeled by the language model and the acoustic model builds and trains the HMMs. In this project it creates a bi-gram language model. This was built using the HTK language modeling tools such as *LNewmap*, *LGPrep*, *LGCopy* and *LBuilt*.

By using the resulted bi-gram model a word lattice is created using the *HBuild* tool. This tool is used to convert input files that represent language models in a number of different formats and output a standard HTK lattice. The main purpose of *HBuild* is to allow the expansion of HTK multi-level lattices and the conversion of bigram language models into lattice format.

4.3 Training

The major process of building the ASR is building and training the acoustic model. The first step of this process was to create a prototype HMM. This prototype defines the structure and the overall form of the set of HMMs. Here a 3-state left-right topology was used to model the HMMs. The second step is to initialize the monophone HMMs. For this purpose HTK uses the *HCompV* tool. Inputs for this tool are the prototype HMM definition and the training data. *HCompV* reads the both inputs and outputs a new definition in which, every mean and covariance is equal to the global speech mean and covariance. So, every state of a monophone HMM gets the same global mean and covariance. Next a Master Macro File (MMF) called *hmmdefs* containing a copy for each of the required monophone HMMs is constructed. The next step is to re-estimate the stored monophones using the embedded re-estimation tool *HERest*. This process estimates the parameters of monophone HMMs from the training set that are intended to model. This is the process of training HMMs. The re-estimation procedure is repeated three times for each of the HMM to train.

After re-estimating the context independent monophone HMMs, we move onto context dependent triphone HMMs. These triphones are made simply by cloning the monophones and then re-estimating using triphone transcriptions. The next step is to re-estimate the new triphone HMM set using the *HERest* tool. This is done in the same way as the monophone HMMs were estimated by replacing the monophone list and the monophone transcription with the corresponding triphone list and the triphone transcription. This process is also repeated three times. The last step in the model building process is to tie states within triphone sets in order to share data and thus be able to make robust parameter estimates. However, the choice of which states to tie requires a bit more subtlety since the performance of the recognizer depends crucially on how accurate the state output distributions capture the statistics of the speech data. In this project it uses decision trees to tie the states within the triphone sets.

The final step of the acoustic modeling is the re-estimation of created tied state triphones and this process is also same as the earlier use of *HERest*. This is also repeated for three times and the final output is the trained acoustic model. Previously created language model is used to

evaluate the trained acoustic model and the evaluation process will be described in the next section.

5 Testing & Evaluation

5.1 Performance Testing

As mentioned in the previous chapter in the data collection process, 106 distinct Sinhala utterances were recorded for the purpose of testing. Before starting the test, the acoustic data files and word level transcriptions were generated for the test speech set. Acoustic data files (files containing extracted speech features from the test set) were generated by executing the *HCopy* tool.

Next, the acoustic data of the test set was input to the built system and recognized using the Viterbi decoding algorithm. In HTK this is done by executing the *HVite* tool. The inputs to the *HVite* tool are, the trained acoustic model, coded acoustic data of the test set, language model word lattice and the pronunciation dictionary.

After generating these files the performance can be measured by comparing the manually created transcription file (file which containing the transcriptions of the input utterances) and the *HVite* generated output transcription file (file containing transcriptions of the recognized utterances). The computation of accuracy rate is done by executing the *HResults* tool. The above computation gives following results:

The percentage of 100% correctly identified sentences was 75.74% (i.e. 80 sentences out of 106 were perfectly correctly recognized). The percentage of correctly identified words in the whole set of test sentences was 96.14%. That is 797 words out of 829 words contained in the 106 sentences were correctly recognized.

5.2 Error Analysis

According to the above results an error analysis was done to identify the causes for the incorrect recognitions. When the incorrectly identified utterances were manually compared with the correct utterances, on most of the utterances only one or two syllables happened to be incorrectly identified. Very few utterances were incorrectly recognized due to incorrect word boundary detections. Only very few utterances were completely incorrectly recognized (as different words).

- Number of incorrectly identified utterances with one syllable changes = 17

- Number of incorrectly identified utterances due to incorrect word boundaries = 7
- Number of incorrectly identified utterances due to completely different words = 4

6 Conclusion

This paper describes an attempt to build an ASR system for continuous Sinhala speech. This section discusses the successfulness of the research, drawbacks and possible future works to improve the work carried out by this research.

6.1 Success & Drawbacks

The primary objective of this project was to build a prototype for a continuous Sinhala speech recognizer. As we were in a very early stage of building ASR systems for continuous speech, it can be said that the primary goal of using open source tools for building a recognizer for Sinhala speech has been achieved to a considerable and sufficient extent. The test results show that the system achieves 75% sentence recognition accuracy and 96% word recognition accuracy (or a word-error rate of just 4%). According to the error analysis it shows that most of the incorrectly identified utterances differed from the correct utterances only by one or two syllables. A better n-gram based language model could potentially help reduce such error further.

The system was trained only from a single female voice. Hence the above results were accurate only for the trained voice. The system gives a very low recognition rate for other human voices. This has to be solved by training the system using a variety of human voices of both male and female. Such an exercise is currently underway at the Language Technology Research Laboratory of the UCSC.

Another goal of this project was to build the system for an unrestricted vocabulary. The Sinhala language has a very large vocabulary in terms of its morphological and phonological productivity. We tried to achieve this goal by building the system using a sample of written Sinhala vocabulary. This vocabulary needs to be extended by adding words to the pronunciation dictionary and adjusting the language model according to it.

6.2 Future Work

The trained model can be improved to build a speaker independent speech recognition system by training the system using a large speech corpus representing voices from various kinds of

human voices. To gain this target the speech corpus should consist of not only male and female human voices, but also should be representative in respect age group, education levels and regions.

Although speech recognition systems built for one language thus far cannot be used to recognize other languages, this research found that there is a large overlap between diverse languages at the phoneme level. Only a few phonemes of Sinhala differed from those of English. However, at the tri-phone level, the inter-dependence of phones with each other can be quite diverse between languages as well as different speakers of the same language. These features are being exploited by newer initiatives that have attempted to build 'universal' speech recognition systems.

Acknowledgments

Authors of this paper acknowledge the support of the members of the Language Technology Research Laboratory of the University of Colombo of School of Computing in conducting this research. Authors would also like to acknowledge the feedback given by two unknown reviewers who have helped in improving the quality of the paper. Any remaining shortcomings however are of the authors alone.

References

- Cole, R. Ward, W. and ZUE, V. 1996. *Speech Recognition*.
<http://cslu.cse.ogi.edu/HLTsurvey/ch1node4.html>.
- Davis, K. H. Biddulph, R. and Balashek, S. 1952. *Automatic Recognition of Spoken Digits*. J. Acoust. Soc. Am. Vol.24, No.6. pp.627-642.
- Dudley, H. Riesz, R. R. and Watkins, S. A. 1939. *A Synthetic Speaker*. Journal of the Franklin Institute. Vol.227. pp.739-764.
- Forgie J. W. and Forgie, C. D. 1959. *Results Obtained from a Vowel Recognition Computer Program*. J. Acoust. Soc. Am. Vol.31, No.11, pp.1480-1489.
- Juang, B.H. and Rabiner, L.R. 2005. *Automatic Speech Recognition – A Brief History of the Technology Development*. Elsevier Encyclopedia of Language and Linguistics.
- Juang, B.H. and Rabiner, L.R. 1991. *Hidden Markov Models for Speech Recognition*. Technometrics. Vol. 33, No. 3, pp. 251-272.
- Kemle, K. A. 2001. *An introduction to speech recognition*. Voice Systems Middleware Education. IBM Corporation.

- Microsoft. *Windows Speech Recognition in Windows Vista*.
<http://www.microsoft.com/enable/products/windowsvista/speech.aspx>.
- Microsoft. *Speech Recognition with Windows XP*
http://www.microsoft.com/windowsxp/using/setup/expert/moskowitz_02september23.msp.
- Olson, H. F. and Belar, H. 1956. *Phonetic Typewriter*.
J. Acoust. Soc. Am. Vol.28, No.6, pp.1072-1081.
- Pike, John. 2006. *Automatic Speech Recognition Techniques*.
<http://www.globalsecurity.org/intell/systems/asr-tech.htm>.
- Rabiner, L. and Juang, B. 1986. *An introduction to hidden Markov models*. IEEE ASSP Magazine, vol. 3, pp. 4-16.
- Rosenfeld, R. 2000. *Two decades of statistical language modeling: Where do we go from here?*. Proceedings of the IEEE. vol. 88, pp. 1270–1278.
- Sakai, J. and Doshita, S. 1962. The Phonetic Typewriter. Information Processing 1962. Proc. IFIP Congress, Munich.
- Weerasinghe, A.R. Wasala, A. and Gamage, K. 2005. *A Rule Based Syllabification Algorithm for Sinhala*. Proceedings of 2nd International Joint Conference on Natural Language Processing, Jeju Island, Korea, pp. 438-449.
- Young, S. Evermann, G. Gales, M. Hain, T. Kershaw, D. Liu, X. Moore, G. Odell, J. Ollason, D. Povey, D. Valtchev, V. and Woodland, P. 2006. *The HTK Book*. Cambridge University Engineering Department, pp. 1-14.

Dzongkha Text-to-Speech Synthesis System – Phase II

Dechen Chhoeden, Chungku
Department of Information Technology
and Telecom , Bhutan
{dchhoeden, chungku}@dit.gov.bt

**Ananlada Chotimongkol,
Anocha Rugchatjaroen, Ausdang
Thangthai, Chai Wutiwivatchai**
HLT Laboratory
National Electronics and Computer
Technology Center, Thailand
{ananlada.chotimongkol, anocha.rug,
ausdang.tha, chai.wut} @nectec.or.th

Abstract

This paper describes the development of advanced Dzongkha text-to-speech (TTS) system which is a marked improvement over the first Dzongkha TTS prototype (Sherpa et al., 2008), using the Hidden Markov Model-based speech synthesis (HTS) method. Advanced Natural Language Processing techniques like word segmentation and phrase boundary prediction were integrated with the earlier prototype to improve the quality of the synthesized speech. These advanced techniques and the integration procedure are explained in this paper. With the inclusion of these advanced modules, we could improve the quality of the synthesized speech as measured by a subjective listening test, Mean Opinion Score (MOS), from 2.41 to 2.98. The procedure of the integration is explained in this paper.

1. Introduction

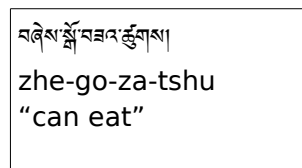
The initial work on TTS focused more on the linguistic aspect. In that, defining a phonetic set for the language, a grapheme-to-phoneme conversion module using a simple phoneme look-up dictionary, text normalization, and collection of speech corpus were some of the main natural language processing (NLP) tasks that were explored. All these modules were put together with a simple text processing model and a speech synthesizer trained using the HTS framework. As the prototype included segmentation only at syllabic level and as word segmentation and phrase boundary prediction which are integral components of natural speech were not taken into account, the output speech sounded robotic as the prosodic information was limited to just the syllable boundaries.

In the second phase of Dzongkha TTS development, advanced NLP techniques have been incorporated to improve the quality of the synthesized speech. This is done by adding a prosody generation module for proper phone duration and pausing which requires development of many sophisticated NLP algorithms including a word segmentation algorithm, a Part-Of-Speech (POS) tagger, a phone duration predicting algorithm and a phrase boundary predictor. These

algorithms require a deeper understanding of the language in order to annotate both speech and text corpora with POS tags, word and phrase boundaries in order to train various statistical models.

1.2. Structure of Dzongkha Text

Dzongkha language belongs to the Sino-Tibetan family of languages. The language structure is not very complicated. It follows the subject-object-verb order of sentence structure like most Asian languages. The smallest unit of text that is meaningful is the 'syllable'. Dzongkha text has syllabic and sentence boundaries; however, there are no spaces between words. A syllable is separated from another syllable by a character ' ' called 'Tsheg' but we do not have word boundaries that separates words from each other. A word is at least one syllable long. As such there are no separate punctuation marks to differentiate the words from one another.



མཚོ་གོ་མཚོ་མཚོ་མཚོ་
zhe-go-za-tshu
"can eat"

Figure 1. Example Sentence

Similarly, though the sentences are clearly delimited by sentence boundaries ('!'), no explicit rules are in place for having phrase boundaries. Spaces inserted between phrases depend on the writer or the speaker. It is necessary for the long sentences to be divided into a number of shorter utterances for the listener or the speaker to be able to understand or read the sentences, respectively. In the case of speech, the boundaries to be inserted can be short pauses between the utterances.

The objective of any TTS system is to emulate natural speech as much as possible. To do that the inclusion of NLP processes of word segmentation and phrase boundary prediction becomes imperative. Hence these processes are integrated in the second phase of Dzongkha TTS development to improve the system.

2. Design and Development

In phase-I of Dzongkha TTS development, the text analysis was completed to convert the text to intermediate forms of syllables and attach their corresponding phonemic representation. And the speech synthesizer would generate speech using HTS method. This model was chosen owing to its small footprint, stability, smoothness and speaker adaptability (Sherpa et al., 2008). Using HMM method, spoken signal is generated from acoustic parameters that are synthesized from context dependent HMM models. In phase-II the same process is followed with the integration of additional NLP modules.

In phase-II, to improve the synthesized speech quality, we have increased the size of the speech database used for training the speech synthesizer. We have also incorporated a prosody generation module to the system to make the synthesized speech more natural with more appropriate phone duration and pausing. The following diagram (Figure 2) illustrates the process that was used to improve the quality of the synthesized speech in the new system.

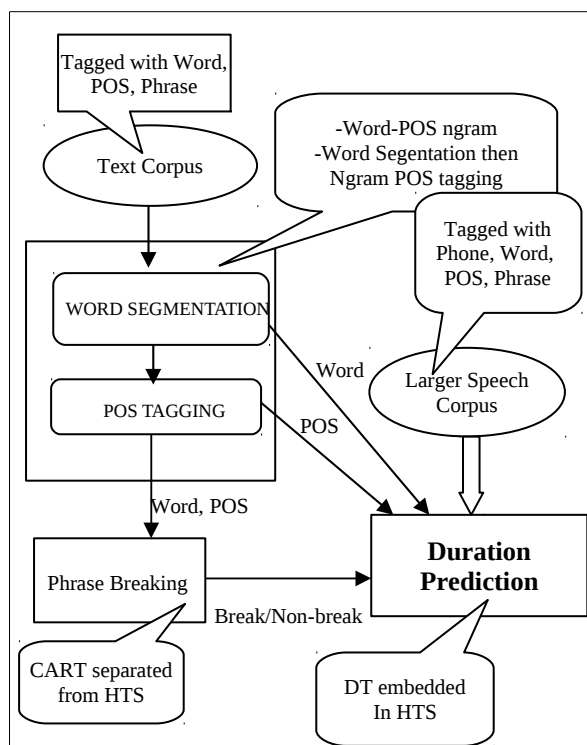


Figure 2. The development process of the Dzongkha TTS system Version II (Training phase)

From the diagram, we can see that word boundary and POS are crucial information for predicting a phrase boundary. Phrase boundaries together with word boundaries and POS tags are, in turn, crucial information for predicting phone duration. Hence, in order to predict phrase boundaries and phone duration in the prosodic generation module, a word

segmentation module and a POS tagger are necessary components for extracting features during the executing phase of the TTS system. The training phase of the Dzongkha TTS system version 2 comprises of 5 steps, data preparation and the development of 4 necessary components: word segmentation, POS tagging, phrase boundary prediction and phone duration prediction.

2.1. Data Preparation

When building a TTS system a number of different types of data needs to be prepared. Text corpus and Speech corpus being the major portion of data, apart from that, data like POS-annotated text are also required. The following describes the data that were prepared:

Text Corpus: Dzongkha Corpus is a collection of text with 600,000 syllables (400,000 words approximately) from different genres like newspaper articles, samples from traditional books, novels, dictionary. A small part of this corpus which contains approximately 40,277 words, was annotated with word boundaries, phrase boundaries and POS tag of each word. These features are necessary for training a phrase boundary prediction and a phone duration prediction modules.

Larger speech corpus: 434 additional sentences were recorded to increase the speech corpus to 943 speech utterances. These sentences covering all the phones in the language, were selected from the text corpus with the selection criteria described by Wutiwivatchai et al. (2007) and Sherpa et al. (2008). A female native speaker recorded the sentences and the resulting wave files were converted to a 44 KHz and 16 bits format. The speaker was the only one available in that particular circumstance.

Increase the syllable entries in the Pronunciation dictionary: About 900 new syllables were transcribed with their respective pronunciation (a sequence of phones). Transcription of syllables involves marking each phone in the syllable with special symbols borrowed from the IPA system.

For example, the syllable 'ཁྱ' is transcribed as 'kh-a-b-0', which consists of three phonemes (initial consonant, vowel and final consonant) and '0' indicates normal tone. High tone is represented by '1'.

Additional features of speech: Word and phrase boundaries were added using special symbols in each sentence (934 sentences from the speech corpus) and each of the words in those sentences were tagged with their respective POS tags. This is a crucial step in our TTS building phase as this data will be used in the "labelling" process of HTS to train HTS model.

Example sentence: མདའ་བརྒྱབ་དེ་མདའ་གཉིས་མ་རང་བཀག་ཟེག་པོ་གཉིས་ཀྱི་

Annotation: d-a-x-0|*/NN/c-a-b-0|*/VBA/d-a-x-0|*/TM/d-a-x-0|*/NN/ny-i-p-1|ch-a-x-0|*/NN/r-a-ng-0|*/CC/k-a-x-0|r-e-x-0|*/NN/ph-o-x-0|n-i-x-0|*/VB/\$-\$-\$-\$|*///
i

'*' indicates the word boundary while “\$-\$-\$-\$|*” is used to represent phrase boundaries.

Preparation of label files for the new sentences:
Preparation of label files with word boundary, phrase boundary and POS information. The data from the above step will be used to prepare label files for the “labelling” process.

2.2. Advanced NLP modules

Word Segmentation

Segmentation of words is basically the process of tokenizing a string of text into words. Many different methods are available for word segmentation nowadays. The accuracy of each of the methods will differ from language to language. Hence a language needs to adopt the best algorithm that gives the best accuracy. For Dzongkha TTS we have adopted the “longest string matching” method for segmenting words, as it is suitable for a small corpus as is our case. From the limited amount of training data, we used this technique which is a dictionary-based method of word segmentation. It depends on a match between word entries in the dictionary and a string of characters in an input text, and segments the text accordingly. Given the input text the algorithm scans the characters from left to right and finds the longest matching entry in the dictionary.

There are many draw-backs with this method. For one there is the problem of unknown word or out-of-vocabulary (OOV) word where some words may not match any word in the dictionary. Secondly, there may be more than one ways to segment the input sequence of characters. These drawbacks can be improved with the adoption of algorithms with better accuracy.

POS Tagging

An integral part of prosody is the POS of the words. An annotated text with POS information for each word, is required to add additional functionality to the TTS system, to support vital features of natural speech. An automatic POS tagger (Chungku, et al., 2010) was used to tag text corpus for phrase boundary prediction.

Example of Dzongkha POS-tagger output text:

Input Text	Output Text
མཱ་པོ་	མཱ་པོ་ NNP
རིན་ལྗན་ལྗན་	རིན་ལྗན་ལྗན་ NNP
༡༩༩༩	༡༩༩༩ ND
ལྷ་	ལྷ་ PP
	PUN

Table 1. Output of POS Tagger

Also the 934 sentence utterances had to be manually POS-tagged along with the word and phrase boundary annotations for training HTS, as explained in Section 2.1.

Example of POS Annotation of raw text:

Raw text: འབྲུག་དང་བཟོ་རྒྱུ་གཉིས་ཀྱི་བར་ན་ ཟུང་འབྲེལ།
POS Annotated equivalent: འབྲུག་ /NNP དང་/CC བར་
 རྒྱུ་ /NNP
 བར་ན་ /CD ཀྱི་/CG ཟུང་འབྲེལ་/NN |PUN

Phrase boundary prediction

Phrases are also an integral part of prosody in speech. A Phrase boundary predictor is necessary in a TTS system for the synthesized speech to sound natural. It will break down a long utterance into meaningful parts. In Dzongkha there are no explicit phrase boundaries, although boundaries at syllabic and sentence levels are present. At every sentence end, it is certain that a speaker must pause and then start reading the next sentence, however it is unsure of the pattern of phrases and pauses within a sentence. In Dzongkha speech, phrase breaks can occur between words, sentences, numbers, spaces, punctuation and symbols.

To predict phrase breaks in a Dzongkha sentence, we use a machine learning algorithm called Classification And Regression Tree (CART). CART is a learning algorithm that is based on a binary decision tree and has been applied widely in the task of phrase boundary prediction (Hansakunbuntheung et al., 2005). Its ability to manipulate both symbolic and real-value features and the ability to handle sparse training data are the advantages of this model.

We use POS information, features at word and syllable levels to train a CART tree to predict whether to insert a pause at the current location or juncture with respect to each word. These features are described in more detail in the following table. The last two rows in the table are the output of the CART decision tree.

Feature	Description
POSL2,P OSL1,PO SR1, POSR2	POS with respect to the position of the current juncture. For e.g, POSL1 is the POS of the word one place on the left hand side of the current juncture. POSR1 is the POS of the word one place to the right with respect to the position of the current juncture. [each]
CurrSylIn Phr	No. of Syllables between the current juncture and the beginning of the current sentence.
CurrWrdI nPhr	No. of Words between the current juncture and the beginning of the current sentence.
WrdInPhr	Total no. of words in the current sentence.
SylInPhr	Total no. of syllables in the current sentence.
NB	Non-break for junctures that is not a phrase break.
B	Break for junctures that is a phrase break.

Table 2. Required Features for each word to train CART

For training CART, the data-set or corpus without removing any punctuation, symbols and numbers were used considering the fact that in real life all kinds of text-data will be present. The required features are extracted from this corpus to train CART, as a result a 'decision tree' is formed which can then be used to automatically predict phrase boundaries for a given input text. It basically looks at the Break (B) or Non-break (NB) status of each word in the training corpus, and puts these information in the decision tree along with other features. In the executing phase of the TTS, the 'B' and 'NB' features of an input string will be output by the "phrase prediction" process, and this output file will be used in the "labelling" process for training HTS to help produce synthesized speech containing silences wherever the phrase breaks were predicted.

Phone duration prediction

To predict the duration of each phone, features like word boundaries, phrase boundaries and POS of each word are necessary, together with phone duration from natural speech. To do that we needed to manually tag the word boundaries, POS tags and the phrase boundaries of the 943 sentence utterances from the speech corpus. In the HTS training process these

information will be utilized to generate a phone duration model.

2.3. HTS Training Process

As in phase I of Dzongkha TTS development, in phase-II as well, HTS speech engine is used as the speech synthesizer of the system. Here the naturalness is improved by adding more context to each phone label for creating duration trees and using more appropriate context-based HMMs. The extra features of POS, word and phrase boundaries, position in word and phrase, are added to the question file (clustering tree) for training additional phrase boundary prediction module. After the training process is successfully completed, HMMs and tree files which are needed for the HTS synthesizer are created. To synthesize a speech file, the synthesizer also needs a context-based phone label file of target speech, generated during the execution process of the system.

2.4. Execution phase: Putting it all together

In the execution phase (Figure 3) all the modules that were developed are integrated together with the HTS, using a command prompt application. To explain the execution process, given a text input, the string of text is first segmented into words. These words are then tagged with their corresponding POS tags. Numbers and dates are normalized into letter-form and the phoneme string for each word is also looked-up using the syllable dictionary (Sherpa, et al., 2008). Phrase breaking process then predicts phrase breaks and non-breaks. After these steps we have an output in the format as shown below. This output file has word entries (from the input text). In that each word has its corresponding POS information, break or non-break situation, and the transcription (letter to sound) of each word along with the tone (0 or 1) of the word.

Word	POS	B/NB	Transcription
ཉིན་མོ་	NN	NB	ny-i-m-0
ལྔ	NN	NB	ny-i-x-1 p-a-x-0
ལྷག་ལེ་ལེ་	NN	NB	t-a-x-0 l-a-x-0 kh-a-x-0
བྱུང་ལྷག་ན	NN	B	j-a-x-0 ph-u-x-0 n-a-x-0

Table 3: Output file having features of a given text

This output file is then used to obtain a context label file for that particular input string. That is then used to generate the synthesized speech using the HTS

synthesizer that was previously trained in the training process.

In this way the TTS Version-2 is capable of producing synthesized speech from a string of input text. The quality of the speech is discussed in the following section.

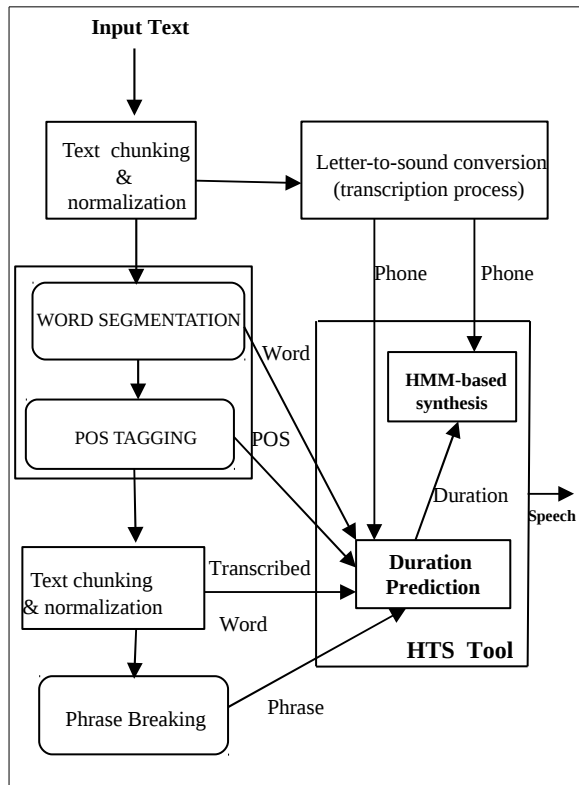


Figure 3. A diagram describing the Execution Phase of the TTS system

3. Evaluation results and Discussions

3.1. Word Segmentation

F-measure was used to evaluate the overall performance of the word segmentation algorithm. The F-measure is calculated using the equations as given below.

F-measure = 2 (Precision X Recall)/(Precision + Recall)	Corr = The number of words in the output that is correctly segmented; RefWord = The total number of words in the reference; OutputWord = The total number of words in the output.
Precision = Corr / OutputWord	
Recall = Corr / RefWord	

Table 4. Equations for finding accuracy of the longest matching algorithm

To evaluate the performance of the longest matching algorithm using Dzongkha text, a dictionary was created from all unique words in the text corpus. The dictionary contains 30,856 words. The test set is the transcript of 943 wave files in the speech corpus also described in Section 2.1. The test set contains 8,701 words. The result from the evaluation is presented in the following table.

Precision	86.73
Recall	84.67
F-measure	85.69

Table 5. Accuracy of the longest matching algorithm

Many different word-segmentation algorithms can be explored to achieve better accuracy for word-segmentation, in future.

3.2. POS Tagging

For the POS tagger, a training data of 20,000 words which were manually tagged with their respective POS tags. This corpus was used to create a working POS tagger. 5000 words were automatically tagged with this tagger. The output from the tagger was manually corrected and found that the accuracy of the tagger was around 80%. This initial tagger was used in the development of the Dzongkha TTS system version-2.

3.3. Phrase boundary prediction

In training CART, 90% of the corpus was used as training data while 10 % of the corpus was used as test data. The training data (train.txt) is a dataset of 37537 words (9022 phrases and 2743 sentences) and the test data (test.txt) is a training data subset containing 3719 words. The performance of a phrase boundary predictor is measured in terms of percentage of correctly predicted breaks and non-breaks, similar to the criteria discussed by Hansakunbuntheung et al. (2005). An accuracy of 88.42% was achieved using precision and recall.

3.3. Evaluation of the system

The system was evaluated using Mean Opinion Scoring method (Viswanathan, 2005). Fifteen native speakers were requested to rate the new synthesized speech (*syn II*), the natural speech and the synthesized speech (*syn I*) produced by the first TTS system, based on the quality and naturalness of the speech samples. The ratings of 1 to 5, 1 being the worst and 5 being the best, were used for the fifteen speech

samples which were jumbled between natural and synthesized samples. After evaluation, recorded speech had an average rating of 2.41 for “syn I” and 2.98 for “syn II”. While the natural speech, understandably, had the highest average score of 4.63, the resulting MOS scores clearly indicate that the new system has improved considerably in comparison to the previous system.

The resulting speech from the system has vastly improved compared to the almost robot like speech output by the earlier system. Still the system has room for improvement by increasing both the text and speech corpus and improving the accuracy of the different modules presented in the paper.

4. Conclusion

This paper presented the integration of advanced NLP modules in Dzongkh TTS system, which included word segmentation, phrase boundary prediction, POS tagging and phone duration prediction. As a result the system has improved. This means that the synthesized speech produced by the system is closer to spoken speech containing word boundaries and some pauses in between phrases within the sentences. The evaluation score based on a subjective listening test was indicative of the fact that the system has improved, yet, it is far from being as good as the natural speech. Future work such as improving the accuracy of the integrated modules, and increasing both the text and speech corpus may help improve the system furthermore.

Acknowledgement

Firstly we would like to thank the donors, the PAN Localization project Secretariat for their continuous support. Again phase-II of Dzongkha TTS would not have progressed as it has without the kind helping hand extended by our friends from Human Language Technology (HLT) lab at NECTEC. We are very much grateful for their kind support and contribution towards the development of the advanced TTS system.

References

Chai Wutiwiwatchai, Anocha Rugchatjaroen, and Sittipong Saychum. 2007. An Intensive Design of a Thai Speech Synthesis Corpus. *The Seventh International Symposium on Natural Language Processing*. Thailand.

Chatchawarn Hansakunbuntheung, Ausdang Thangthai, Chai Wutiwiwatchai, and Rungkarn Siricharoenchai. 2005. Learning methods and features for corpus-based phrase break prediction on Thai. *European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1969-1972.

Choochart Haruechaiyasak, Sarawoot Kongyoung, and Matthew N. Dailey. 2008. A Comparative Study on Thai Word Segmentation Approaches. *In Proceedings of ECTI-CON*. Krabi, Thailand.

Chungku, Gertrud Faaß, and Jurmey Rabgay. 2010. Building NLP resources for Dzongkha: A tagset and a tagged corpus. *Proceedings of the Eighth Workshop of Asian Language Resources (WS1), 23rd International Conference on Linguistics (Coling 2010)*, 103-110. Beijing, China,.

Dechen Chhoeden, Chungku, Anocha Rugchatjaroen, Ausdang Thangthai, Chai Wutiwiwatchai, and Ananlada Chotimongkol. 2010. Technical report on Dzongkha Speech Synthesis System – Phase II.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, 44-49. Manchester, UK.

Mahesh Viswanathan. 2005. Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (MOS) scale. *Computer, Speech and Language*, volume 19, 55-83.

Uden Sherpa, Dawa Pemo, Dechen Chhoeden, Anocha Rugchatjaroen, Ausdang Thangthai, and Chai Wutiwiwatchai. 2008. Pioneering Dzongkha text-to-speech synthesis. *Proceedings of the Oriental COCOSDA*, 150–154. Kyoto, Japan.

Bangla Text to Speech using Festival

Firoj Alam

Center for Research on Bangla Language Processing
BRAC University.
firojalam@gmail.com

S.M. Murtoza Habib

Center for Research on Bangla Language Processing
BRAC University.
murtoza@gmail.com

Mumit Khan

Center for Research on Bangla Language Processing and
Department of Computer Science and Engineering
BRAC University.
mumit@bracu.ac.bd

Abstract

This paper describes the development of the first, usable, open source and freely available Bangla Text to Speech (TTS) system for Bangladeshi Bangla using the open source Festival TTS engine. Besides that, this paper also discusses a few practical applications that use this system. This system is developed using diphone concatenation approach in its waveform generation phase. Construction of a diphone database and implementation of the natural language processing modules are described. Natural language processing modules include text processing, tokenizing and grapheme to phoneme (G2P) conversion that were written in Festival's format. Finally, a test was conducted to evaluate the intelligibility of the synthesized speech.

Index Terms: speech synthesis, diphone

1 Introduction

Bangla (exonym: Bengali) is one of the most widely spoken languages of the world (it is ranked between four¹ and seven² based on the number of speakers), with nearly 200 million native speakers. However, this is one of the most under-resourced languages which lack speech applications. The aim of this project is to develop a freely available Bangla text to speech system. A freely available and open-source TTS system for Bangla language can greatly aid human-computer interaction: the possibilities are endless – such a system can help overcome the literacy barrier of the common masses, empower the visually impaired population, increase the possibilities of improved man-machine interaction through on-line newspaper reading from the in-

ternet and enhancing other information systems. A touch screen based kiosk that integrates a Bangla TTS has the potential to empower the 49% of the population who are illiterate³. A screen reader that integrates a Bangla TTS will do the same for the estimated 100 thousand visually impaired citizens of Bangladesh. A Text to Speech is a computer based system capable of converting computer readable text into speech. There are two main components such as Natural Language Processing (NLP) and Digital Signal Processing (DSP) [Thierry 1997][Paul 2009][A.W. Black et al. 2003]. The NLP component includes pre-processing, sentence splitting, tokenization, text analysis, homograph resolution, parsing, pronunciation, stress, syllabification and prosody prediction. Working with pronunciation, stress, syllabification and prosody prediction sometime is termed as linguistic analysis. Whereas, the DSP component includes segment list generation, speech decoding, prosody matching, segment concatenation and signal synthesis. Pre-processing is the process of identifying the text genre, character encoding issues and multilingual issues. Sentence splitting is the process of segmenting the document text into a list of sentences. Segmenting each sentence into a list of possible tokens can be done by tokenization. In text analysis part, different semiotic classes were identified, and then using a parser each token is assigned to a specific semiotic class. After that, verbalization is performed on non-natural language token. Homograph resolution is the process of identifying the correct underlying word for ambiguous token. The process of generating pronunciation from orthographic representation can be done by pronunciation lexicon and grapheme-to-phoneme (G2P) algorithm. Prosody prediction is the process of identifying the phrase break, prominence and intonation

¹<http://www2.ignatius.edu/faculty/turner/languages.htm>, Last accessed December 26, 2007.

²http://en.wikipedia.org/wiki/List_of_languages_by_total_speakers, Last accessed December 26, 2007.

³ Bangladesh Bureau of Statistics, 2004. http://www.bbs.gov.bd/dataindex/stat_bangladesh.pdf

tune. There has not been much work done on prosody in this paper. DSP component or waveform generation is the final stage of a TTS. This involves the production of acoustic signals using a particular synthesis approaches such as formant synthesis, articulatory synthesis and concatenation based synthesis. The attempt that has been made here is the second generation diphone concatenation based synthesis, using widely usable Festival framework [A.W. Black et al. 2003].

2 Literature survey

Significant effort has been made for different languages to develop TTS using the Festival framework such as English, Japanese [A.W. Black et al. 2003], Welsh [R.J. Jones et al. 2006], Telugu [C. Kamisetty et al. 2006], [S.P. Kishore et al. 2002], Hindi [S.P. Kishore et al. 2002], [A.G. Ramakishnan et al. 2004], Turkish [Ö. Sallor et al. 2003] and Sinhala [Ruvan et al. 2007]. However, very little work has been done on Bangla. Several attempts have been made in the past, where different aspects of a Bangla TTS system were covered in [Tanuja et al. 2005], [Asok 2002], [Shyamal et al. 2002] and [Aniruddha et al. 2004]. [Tanuja et al. 2005] showed different TTS modules such as optimal text selection, G2P conversion and automatic segmentation with experimental results. Phoneme and partname were used to develop voice database and ESNOLA technique were used for concatenation [Asok 2002], but the quality of the system suffers due to the lack of smoothness. Shyamal et al. [Shyamal et al. 2002] showed some practical applications with Bangla TTS system using ESNOLA technique. In [Aniruddha et al. 2004] author showed the pronunciation rule and phoneme to speech synthesizer using formant synthesis technique. Another attempt has been made to develop Bangla TTS using multisyn unit selection and unit selection technique within Festival framework in [Firoj et al. 2007] but the system was developed on a limited domain and could not be used as a general purpose TTS system. To the best of our knowledge this is the first complete work for general purpose, open source, freely available and platform independent Bangla Text to Speech system for Bangladeshi Bangla.

3 Development

3.1 Bangla writing system

Bangla is written left to right in horizontal lines with a left-to-right heatstroke (is called matra).

The presence and absence of heatstroke has significant implications to distinguish consonant conjunct, dependent and independent vowel. Words are delimited by a space in general. Vowels have corresponding full-character forms when they appear in an absolute initial position of a word. Generally a vowel followed by a consonant takes a modified shape and placed at the left, right or both, or at the bottom of the consonant which are signifies as vowel modifiers. The inventory of Bengali script is made up of 11 vowels, 39 consonants, 216 consonant conjuncts, 10 digits, modifiers, punctuation marks and a few symbols [Bangal Academy 1992][Wikipedia 2010]. The vowel and consonant characters are called basic characters. The consonant conjunct is joined by ‘hasanta’. The concept of upper and lower case is absent in Bangla script. English equivalent full stop is the Bengali punctuation mark the down-stroke dari (।); Unicode - \u0964. Commas, semicolons, colons, quotation marks, etc. are the same as in English.

3.2 Bangla phoneme inventory

The phoneme inventory of Bangla consists of 30 consonants, 14 monophthong vowels (oral and nasal vowels) and 21 diphthongs [Firoj et al. 2008 (b)] [Firoj et al. 2008 (a)]. Consonants and vowels are shown in Table 1 and Table 2. The diphthongs are the following: অও/ao/, আই/ai/, আউ/au/, আয়া/aja/, ইউ/iu/, ইএ-ইয়ে/ie/, ইও/io/, ইয়া-ইআ/ia/, উই/ui/, উয়া-উআ/ua/, উয়ে/ue/, উয়ো-উও/uo/, এই/ei/, এউ/eu/, এও/eo/, এয়া-এআ/ea/, এয়া/æa/, ওই/oi/, ওউ/ou/, ওয়া-ওআ /oa/, ওয়ে/oe/.

	Front	Central	Back
High	ই/i, ই̃/ī,		উ/u, উ̃/ū/
High-Mid	এ/e, এ̃/ē		ও/o, ও̃/ō
Mid-Low	এয়া/æ, এয়া̃/æ̃		অ/a, অ̃/ā/
Low		আ/a, আ̃/ā	

Table 1: Vowel phoneme inventory

3.3 Natural language processing in Festival

Festvox [A.W. Black et al. 2003] provides different natural language processing modules for building a new voice. These modules can be generated automatically which appears as a form of scheme files. The scheme files need to be customized for a new language. The language spe-

cific scripts (phone, lexicon and tokenization) and speaker specific scripts (duration and intonation) can be externally configured and implemented without recompiling the system [A.W. Black et al. 2003]. Since the templates are scheme files, which is typically an interpreted language, so recompilation is not required. The following NLP related tasks are involved when building a new voice in Festvox:

- Defining the phoneset
- Tokenization and text normalization
- Pronunciation: Lexicon and grapheme to phoneme conversion.
- Implementation of syllabification and stress
- Prosody: Phrase breaking, accent prediction, assignment of duration to phones and generation of f0 contour.

whitespace and the punctuation marks, which is used in our implementation to tokenize Bangla text. After tokenization, text normalization is performed. In text normalization, the first task is to identify the semiotic classes. The following section discusses the semiotic class [Paul 2009] (as opposed to say NSW) identification, tokenization and standard word generation and disambiguation rule. Moreover, this work has been done separately before implementing into Festival.

We identified a set of semiotic classes which belong to the Bangla language. To do this, we have selected a news corpus [Prothom-Alo 2009] [Khair et al. 2006] with 18100378 tokens and 384048 token types [Khair et al. 2006], forum [forum.amaderprojukti.com 2008] and blog [www.somewhereinblog.net 2008]. After that we

		Bilabial		Dental		Alveolar		Post- Alveolar		Palatal	Velar		Glottal
Stops	voiceless	প /p/	ফ /p ^h /	ত /t/	থ /t ^h /	ট /t/	ঠ /t ^h /	চ /c/	ছ /c ^h /		ক /k/	খ /k ^h /	
	voiced	ব /b/	ভ /b ^h /	দ /d/	ধ /d ^h /	ড /d/	ঢ /d ^h /	য, জ /j/	ঝ /j ^h /		গ /g/	ঘ /g ^h /	
Nasals		ম /m/				ন, ণ /n/					ং, ঞ /ŋ/		
Trill						র /r/							
Flap						ড়, ঢ় /r/							
Fricatives						শ, স /s/				ষ, ষ, স /ʃ/			হ, ঙ /h/
Lateral						ল /l/							
Approximant										য় /j/			

Table 2: Consonant phoneme inventory

3.3.1 Defining phoneset for Bangla

The phoneset that has been explained in section 3.2 was implemented in festival format which had to be transcribed into ASCII. Festvox has a separate module to implement this phoneset with their features. For vowel, the features include height, length (e.g: short, long, diphthong and schwa), front vs back, lip rounding, and tense vs lex. For consonant, the features include place of articulation (e.g: bilabial, dental, alveolar, post-alveolar, palatal, velar and glottal), manner of articulation (e.g: stop, nasal, trill, flap, fricative, lateral, glide/approximant), aspiration and voicing.

3.3.2 Text analysis and text normalization

Like any conventional writing system, Bangla script uses whitespace as a delimiter which helped us to make a one-to-one mapping between tokens and words. Besides whitespace, Bangla script also uses punctuation (such as darsi, ?, !, ;) as a delimiter. The default text tokenization methodology available in Festival is the

proceeded in two steps to identify the semiotic classes. Firstly, a python [Python 2008] script was used to identify the semiotic class from news corpus and manually checked the semiotic classes in the corpus of forum and blog. Secondly, we defined a set of rules according to context of homographs or ambiguous tokens to find the semiotic classes. The resulted set of semiotic classes of Bangla text is shown in Table 3.

Semiotic class/token type	Example
English text	জাভা Platform Independent বলে
Bangla text	আমি বাংলায় কথা বলি
Numbers (cardinal, ordinal, roman, floating number, fraction, ratio)	১২১, ২৩, ২৩৪; ১ম, ২য়, ৩য়; I, II, III, ১২.২৩, ২৩/৩৩.৩৩; ১/২, ২৩/২৩; ১২:১২
Telephone and mobile number	০২৯৫৬৭৪৪৭; ০১৫২৩০৩৩৯৮ (19 different formats)
Years	২০০৬; ১৯৯৮; ৯৮ সালে
Date	০২-০৬-২০০৬ (12 different formats)

Time	৪.২০ মিঃ; ৪.২০ মিনিট;
Percentage	১২%
Money	১০ ট
E-mail	আমার ই-মেইল ঠিকানা: abc@yahoo.com
URL	সফটওয়্যারটি http://googlecode.com সাইট
Abbreviation	ডঃ; মোঃ; সাঃ
Acronym	ঢাবি; বাউবি, কেবি
Mathematical equation	(১+২=৩)

Table 3: Possible token type in Bangla text

A set of tags defined for each semiotic class and assigned these tags to each class of tokens. The tokenization undergoes three levels such as: i. Tokenizer ii. Splitter and iii. Classifier. Whitespace is used to tokenize a string of characters into a separate token. Punctuations and delimiters were identified and used by the splitter to classify the token. Context sensitive rules were written as whitespace is not a valid delimiter for tokenizing phone numbers, year, time and floating point numbers. Finally, the classifier classifies the token by looking at the contextual rule. For each type of token, regular expression were written in festival scheme.

The token expander expands the token by verbalizing and disambiguating the ambiguous token. Verbalization [Paul 2009] or standard word generation is the process of converting non-natural language text into standard words or natural language text. A template based approach [Paul 2009] such as the lexicon was used for number cardinal, ordinal, acronym, and abbreviations. Abbreviations are productive and a new one may appear, so an automatic process may require solving unknown abbreviations. In case of Bangla acronyms, most of the time people say the acronym as a full form without expanding it. For example, দুদক /dudok/ expands to দুর্নীতি দমন কমিশন /durniti dmon komishon/ but people say it as দুদক /dudok/. Bangla has the same type of non-natural language ambiguity like Hindi [K. Panchapagesan et al. 2004] in the token year-number and time-floating number. For example: (i). the token ১৯৯৮ (1998) could be considered as a year and at the same time it could be considered as number and (ii). the token ১২.৮০ (12.80) could be considered as a floating point number and it could be considered as a time.

Context dependent hand written rules were applied for these ambiguities. In case of Bangla, after time pattern ১২. ৩০ (12.30) we have a token মিঃ (minute), so we look at the next token and decide whether it is time or a floating point number. There are rare cases where context dependent rules fail in year-number ambiguity then we verbalize the token as a pair of two digits. For example, the token ১৯৯৮ (1998), we expand it as উনিশ শত আটানব্বই (Nineteen hundred ninety eight) rather than এক হাজার নয় শত আটানব্বই (one thousand nine hundred ninety eight). The natural language text is relatively straightforward, and Bangla does not have upper and lower case. The system implemented based on the work of [Firoj et al. 2009], claims that the accuracy of the ambiguous token is 87%.

3.3.3 Pronunciation

This system takes the word based on orthographic linguistic representation and generates a phonemic or phonetic description of what is to be spoken by the subsequent phases of TTS. In generating this representation we used a lexicon of known words and a grapheme-to-phoneme (G2P) algorithm to handle proper names and unknown words.

We developed a system lexicon [2, pp215] where the entries contain orthography and pronunciation in IPA. Due to the lack of a digitized offline lexicon for Bangla we had to develop it manually by linguistic experts. To the best of our knowledge this is the first digitized IPA incorporated and syllabified lexicon. The lexicon contains 93K entries where 80K entries entered by hand and the rest of them were automatically generated by G2P system [Ayesha et al. 2006]. The performance of this G2P system is 89.48%. Therefore, the automatically generated entries had to be checked manually to maintain the quality of the lexicon by expert linguists. The system is now available in online for public access [CRBLP 2010]. Another case needs to be handled in order to implement the lexicon into Festival. The Unicode encoded phonetic representation needs to be converted into ASCII to incorporate into festival.

We have implemented the G2P algorithm that is proposed by Ayesha et al. [Ayesha et al. 2006] to handle unknown words and proper name. In Festival, the UTF-8 textual input was converted into ASCII based phonetic representation in a Festival's context sensitive rule [A.W. Black et al. 2003]. The rules were re-written in UTF-8

multi-byte format following the work done for Telugu [C. Kamisetta et al. 2006] and Sinhala [Ruvan et al. 2007]. The method was proven to work well with promising speed. The rules proposed in [Ayesha et al. 2006] were expanded up to 3880 rules when re-written in Festival context sensitive format.

Another attempt has been made to reduce the size of the lexicon for TTS. The lossless compression [Paul 2009] technique was applied to reduce the size of the lexicon. Lossless compression technique is a technique where the output is exactly the same as when the full lexicon is used. It is just the generalities of the lexicon that has been exactly captured in a set of rules. This technique reduces the size of our lexicon to ~50K entries from 93K.

3.3.4 Syllabification and stress

Festival’s default syllabification algorithm based on sonority sequencing principle [A.W. Black et al. 2003] is used to syllabify the Bangla words. Besides the default syllabification algorithm, our lexicon has also been syllabified along with pronunciation.

Little work has been done on Bangla stress. Identifying the stress pattern for Bangla is beyond the scope of this paper. Considering Bangla as a stress less language we have used Festival’s default stress algorithm. In our implementation of lexicon we have not incorporated the stress marker.

3.3.5. Prosody Implementation

Prosody is one of the important factors contributing to natural sounding speech. This includes phrasing, accent/boundary prediction, duration assignment to phones and f0 generation. The presence of phrase breaks in the proper positions of an utterance affects the meaning, naturalness and intelligibility of the speech. Festival supports two methods for predicting phrase breaks. The first one is to define a Classification and Regression Tree (CART). The second and more elaborate method of phrase break prediction is to implement a probabilistic model using probabilities of a break after a word, based on the part of speech of the neighboring words and the previous word [A.W. Black et al. 2003]. However, due to the lack of a POS tagger for Bangla, we have not able to construct a probabilistic model yet. Therefore, we decided to use a simple CART based phrase breaking algorithm described in [A.W. Black et al. 2003]. The algorithm is based on the assumption that phrase boundaries are

more likely between content words and function words. A rule is defined to predict a break if the current word is a content word and the next is seemingly a function word and the current word is more than 5 words from a punctuation symbol. Since function words are limited in a language so we specified them as function words and considered rest of them as content words. The function words that we used here to implement the phrase break model is shown in Table 4. These function words need to be converted into ASCII form to incorporate into festival phrase breaking algorithm.

Function words
অ, অতএব, অথচ, অথবা, অধিকন্তু, অপেক্ষা, অর্থাৎ, আর, আরও, এ, এই, এবং, ও, কিংবা, কিন্তু, তথা, তথাপি, তবু, তবুও, তাই, তো, নইলে, নইলে, নতুবা, নয়তো, না-হয়, বটে, বরং, বরঞ্চ, বস্তুত, বা, যথা, যদি, যদিও, যে, যেন, যেহেতু, সুতরাং, হঠাৎ, হয়তো

Table 4: Function words

To predict accent and boundary tone, Festival uses simple rules to produce sophisticated system. To make a more sophisticated system such as a statistical model one needs to have an appropriate set of data. Due to the lack of the availability of this data we used a simple accent prediction approach [A.W. Black et al. 2003] which proved surprisingly well for English. This approach assigns an accent on lexically stressed syllable in all content words.

Festival uses different approach for F0 generation such as F0 by rule, CART tree and tilt modeling. In our implementation we used rule based approach. An attempt has been made to make a CART tree based model from the data; however, surprisingly that has not been work well. Several duration models support by Festival such as fixed models, simple rules models, complex rules models and trained models. We used fixed duration model that was implemented from the work done by Firoj et al. [Firoj et al. 2008 (b)][Firoj et al. 2008 (a)].

3.4 Development of diphone database

Developing a speech database is always time consuming and laborious. The basic idea of building a diphone database is to explicitly list all phone-phone combination of a language. It is mentioned in section 3.2 that Bangla language has 30 consonants and 35 vowels (monophthong,

diphthong) phonemes. In general, the number of diphone in a language is the square of the number of phones. Since Bangla language consists of 65 phones, so the number of diphones are (65×65) 4225. In addition, silence to phones are (1×65) 65, phones to silence are (65×1) 65 and a silence. So the total number of diphones is 4336. In the first step, a list has been made to maintain all the possible vowel consonant combination with the following pattern: VC, CV, VV, CC, SIL_V, SIL_C, V_SIL, C_SIL and SIL. Here SIL is silence, V is vowel and C is consonant. Silence is considered as a phoneme, usually taken at the beginning and ending of the phonemes to match the silences occurring before, between and after the words. They are therefore an important unit within the diphone inventory. These diphones were embedded with carrier sentences using an external program. The diphone is inserted in the middle of the word of a sentence, minimizing the articulatory effects at the start and end of the word. Also, the use of nonsense words helped the speaker to maintain a neutral prosodic context. Though there have been various techniques to embed diphone with carrier sentences, here nonsense words were used to form carrier sentences [A.W. Black et al. 2003]. In this list, there could be redundant diphones those need to be marked and omitted. The study of phonotactics says that all phone-phone pair cannot be exist in a language. Due to the lack of existing work and linguistic experts we were not able to work on this phenomenon. Therefore, the whole diphone list was selected for recording.

Since speaker choice is perhaps one of the most vital areas for recording so a careful measure had taken. Two potential speakers was chosen and their recording were played to a listening group and asked them which they prefer. According to the measurement of the listening group a male speaker was chosen who is a professional speaker and aged 29.

As far as recording conditions is concerned, we tried to maintain as high quality as possible. The speech data was digitized at a sample rate 44.1 kHz, sample width 24-bit resolution and stored as wave format. After each recording, the moderator checked for any misleading pronunciation during the recording, and if so, the affected utterances were re-recorded.

There were a few challenges in the recording. First, speaker was asked to keep the speaking style consistent. Second, speaker was supervised to keep the same tone in the recording.

The most laborious and painstaking task is to clean the recording and then hand-labeled the diphone using the speech analysis software tool 'Praat'⁴. During labeling, at first we labeled phone boundary, then automatically marked the diphone boundary using Praat script. Another important factor is that, every boundary should be placed in zero crossing. Failing to do so produces audible distortions, this in turns generates clicks. Afterwards, a script was written to transform Praat textgrid files into diphone index file (.est) [A.W. Black et al. 2003] as required by Festival.

Festival, in its publicly distributed form only supports residual excited Linear Predictive Coding (LPC). This method requires pitch marks, LPC parameters and LPC residual values for each diphone in the diphone database. The script *make_pm_wave* provided by speech tools [A.W. Black et al. 2003] was used to extract pitch marks from the wave files. Then, the *make_lpc* command was invoked in order to compute LPC coefficients and residuals from the wave files [A.W. Black et al. 2003]. To maintain an equal power we used proprietary software tool to normalize it in terms of power so that all diphones had an approximately equivalent power. After that the diphone database was grouped in order to make it accessible by Festival's UniSyn synthesizer module, and to make it ready for distribution.

4 Integration with applications

The Bangla Text to Speech runs on Linux, Windows and Mac OSX. There is also a web-enabled front-end for the TTS, making this tool available at anytime and from anywhere.

Since Festival is incapable of reading UTF-8 text files with byte-order marker (BOM) so manual BOM removal patch was used which was written by Weerasinghe et al. [Ruvan et al. 2007]. This patch was incorporated with Festival text processing module.

To develop windows version we had motivated by the work carried out in the Welsh and Irish Speech Processing Resources (WISPR) project [B. Williams et al. 2006]. Following the work of WISPR, we implemented TTS using Microsoft Speech Application Programming Interface (MS-SAPI) which provides the standard speech synthesis and speech recognition interface within Windows applications [Microsoft

⁴ Available from: <http://www.praat.org>

1999]. Consequently, the MS-SAPI compliant Bangla voice is accessible via any speech enabled Windows application. The system has been tested with NVDA⁵ and Dolphin⁶ screen reader. Moreover, it is also tested with WordTalk⁷, a free text-to-speech plug-in for Microsoft Word which runs as a macro. Currently Bengali speaking print disabled community accessing local language content using Bangla Text to speech system via screen reader.

Besides, there are few other applications that currently testing this system such as talking dictionary, DAISY⁸ book, agro-information system and news reader. Using this system one of the newspapers in Bangladesh developed their audio version of newspaper to make mp3 of their daily content.

5 Evaluation

Any real system needs to undergo rigorous testing before deployment. Though TTS testing is not a simple or widely agreed area, it is widely agreed that a TTS system has two main goals on system test; that is a synthesized speech should be i) intelligible and ii) natural. Intelligibility test can be performed by word recognition tests or comprehension tests where listeners are played a few words either in isolation or in a sentence and asked which word(s) they heard. In naturalness test, listeners are played some speech (phrase or sentence) and simply asked to rate what they hear. This can be done by mean opinion score. Since these testing may not always be the best approach so people also use unit testing approach.

As our goal was to make a general-purpose synthesizer, a decision was made to evaluate it under the intelligibility criterion and unit testing on a few components. The most commonly used word recognition test - modified rhyme test (MRT) [Paul 2009] was designed to test Bangla TTS system. Based on the MRT we designed a set of 77 groups - 5 words each. Therefore a set of 385 words came into testing. The words in each group are similar and differ in only one consonant and the users were asked to account which word they have heard on a multiple choice sheet. Based on the test the overall intelligibility of the system from 6 listeners is 96.96%. Besides the

intelligibility test, we have performed a unit test on text normalizer and G2P converter. The performance of text normalizer is 87% only for ambiguous tokens and that of G2P converter is 89%.

6 Conclusions

Here the development of the first-ever complete Text to Speech (TTS) system has described, that can convert a Unicode encoded Bangla text into human speech. It is distributed under an open source license to empower both the user and developer communities. This TTS system can also be used with any available Screen Reader. In addition to the standalone TTS client, it can be integrated into virtually any application, and can also be accessed as a Web Service. Incorporating this technology in various applications such as screen reader for the visually impaired, touch screen based agro-information system, talking books, telecenter applications, e-content, etc., can potentially bridge the literacy divide in Bangladesh, which in turn goes towards bridging the digital divide. An evaluation of the system has been done based on MRT and unit testing on a few components to check intelligibility.

Since the voice developed here is diphone concatenation based and it lacks proper intonation modeling so it produces robotic speech. Therefore, a natural sounding voice needs to be made in future, which could be performed by developing a unit selection voice. Besides that, a few works need to be done in future to improve the intelligibility of the system such as POS tagger, improvement of G2P algorithm, improvement of text normalizer and working on intonation modeling.

Acknowledgments

This work has been supported in part by the PAN Localization Project (www.panl10n.net), grant from the International Development Research Center (IDRC), Ottawa, Canada. We would also like to thank Dr Sarmad Hussain (NUCES), and Naira Khan (Dhaka University).

References

- A.G. Ramakishnan, K. Bali, P.P. Talukdar and N.S. Krishna, 2004, Tools for the Development of a HindiSpeech Synthesis System, In 5th ISCA Speech Synthesis Workshop, Pittsburgh, 2004, pp. 109-114.
- A.W. Black, and K.A. Lenzo, 2003, Building Synthetic Voices, Language Technologies Institute, Carnegie Mellon University and Cepstral LLC. Retrieved from: <http://festvox.org/bsv/>

⁵ NVDA – NonVisual Desktop Access. www.nvda-project.org/

⁶ Dolphin screen reader. www.yourdolphin.com/

⁷ Wordtalk - www.wordtalk.org.uk/

⁸ DAISY - Digital Accessible Information System

- Aniruddha Sen, 2004, Bangla Pronunciation Rules and a Text-to-Speech System, Symposium on Indian Morphology, Phonology & Language Engineering, pp. 39.
- Asok Bandyopadhyay, 2002, Some Important Aspects of Bengali Speech Synthesis System IEMCT Pune, June 24-25 .
- Ayesha Binte Mosaddeque, Naushad UzZaman and Mumit Khan, 2006, Rule based Automated Pronunciation Generator, Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh, December 2006.
- B. Williams, R.J. Jones and I. Uemlianin, 2006, Tools and Resources for Speech Synthesis Arising from a Welsh TTS Project, Fifth Language Resources and Evaluation Conference (LREC), Genoa, Italy, 2006.
- C. Kamisetty and S.M. Adapa, 2006, Telugu Festival Text-to-Speech System, Retrieved from: http://festivalte.sourceforge.net/wiki/Main_Page
- CRBLP, 2010, CRBLP pronunciation lexicon, [Online], Available: <http://crblp.bracu.ac.bd/demo/PL/>
- Firoj Alam, Promila Kanti Nath and Mumit Khan, 2007, Text To Speech for Bangla Language using Festival, Proc. of 1st International Conference on Digital Communications and Computer Applications (DCCA2007), Irbid, Jordan
- Firoj Alam, S. M. Murtoza Habib and Mumit Khan, 2008 (a), Research Report on Acoustic Analysis of Bangla Vowel Inventory, Center for Research on Bangla Language Processing, BRAC University.
- Firoj Alam, S. M. Murtoza Habib and Mumit Khan, 2009, Text Normalization System for Bangla, Conference on Language and Technology 2009 (CLT09), NUCES, Lahore, Pakistan, January 22-24.
- Firoj Alam, S. M. Murtoza Habib and Professor Mumit Khan, 2008 (b), Acoustic Analysis of Bangla Consonants, Proc. Spoken Language Technologies for Under-resourced language (SLTU'08), Vietnam, May 5-7, page 108-113.
- forum.amaderprojukti.com, 2008, Forum
- K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A.G. Ramakrishnan, 2004, Hindi Text Normalization, Fifth International Conference on Knowledge Based Computer Systems (KBCS), Hyderabad, India, 19-22 December 2004. Retrieved (June, 1, 2008).
- Khair Md. Yeasir Arafat Majumder, Md. Zahurul Islam, Naushad UzZaman and Mumit Khan, 2006, Analysis of and Observations from a Bangla News Corpus, in proc. 9th International Conference on Computer and Information Technology, Dhaka, Bangladesh, December.
- Microsoft Corporation.: Microsoft Speech SDK Version 5.1, 1999, Retrieved from: <http://msdn2.microsoft.com/ens/library/ms990097.aspx>.
- Smith, J. O. and Abel, J. S., Bark and ERB Bilinear Trans-forms", IEEE Trans. Speech and Audio Proc., 7(6):697-708.
- Ö. Salor, B. Pellom and M. Demirekler, 2003, Implementation and Evaluation of a Text-to-Speech Synthesis System for Turkish, Proceedings of Eurospeech-Interspeech, Geneva, Switzerland, pp. 1573-1576.
- Paul Taylor, 2009, Text-to-Speech Synthesis, University of Cambridge, February.
- Prothom-Alo, 2009, www.prothom-alo.com, Daily Bengali newspaper
- Python, 2008, www.python.org, version 2.5.2
- R.J. Jones, A. Choy and B. Williams, 2006, Integrating Festival and Windows, InterSpeech 2006, 9th International Conference on Spoken Language Processing, Pittsburgh, USA.
- Ruvan Weerasinghe, Asanka Wasala, Viraj Welgama and Kumudu Gamage, 2007, Festival-si: A Sinhala Text-to-Speech System, Proceedings of Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, Page 472-479.
- S.P. Kishore, R. Sangal and M. Srinivas, 2002, Building Hindi and Telugu Voices using Festvox, Proceedings of the International Conference On Natural Language Processing 2002 (ICON-2002), Mumbai, India.
- Shyamal Kr. DasMandal, Barnali Pal , 2002, Bengali text to speech synthesis system a novel approach for crossing literacy barrier, CSI-YITPA(E)
- Tanuja Sarkar, Venkatesh Keri,. Santhosh M and Kishore Prahallad, 2005, Building Bengali Voice Using Festvox, ICLSI.
- Thierry Dutoit, 1997, An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers
- Wikipedia contributors. Bengali script [Internet]. Wikipedia, 2010, The Free Encyclopedia; 2010 Jul 22, 17:27 UTC [cited 2010 Jul 30], Available online at: http://en.wikipedia.org/w/index.php?title=Bengali_script&oldid=374884413.
- www.somewhereinblog.net, 2008, Blog

A Corpus Linguistics-based Approach for Estimating Arabic Online Content

Anas Tawileh
Systematics Consulting

anas@systematics.ca

Mansour Al Ghamedi
King Abdulaziz City for Science and
Technology

mghamdi@kacst.edu.sa

Abstract

This paper presents the results of a research project for the development of a corpus-based indicator for Arabic online content. The project entailed the construction and analysis of three Arabic language corpora from online and offline sources. These corpora were then utilized in the estimation of the index size of two major search engines. The overlap between these two indices was determined, and an indication of the size of the indexed Arabic web was deduced. The developed indicator suggests that the current size of the indexed Arabic web exceeds 400 million pages.

1 Introduction

In today's interconnected world, the competitive advantages of societies and nations are largely determined by the degree of technology adoption and utilization by these societies. Information and Communication Technologies (ICTs) are considered to be a critical component in the emergence of knowledge societies. The concept of the knowledge society is based on the premises that knowledge can be leveraged to address developmental challenges and improve the quality of life of societies around the world. Mounting evidence supports this argument (Mansell and Wehn, 2000; Stone and Maxwell, 2005), and many countries are launching proactive initiatives to capitalize on knowledge and ICTs for delivering on their development agendas.

It has been suggested that the Arab region is facing particular challenges in its attempts at shifting into knowledge societies. A major challenge

that is frequently cited is the severe shortage of relevant local content in the Arabic language. Researchers and practitioners alike argue that without a critical mass of local content, which constitutes the cornerstone of the knowledge society, Arab countries cannot reap the benefits of the global information revolution.

However, despite the importance of understanding the size and quality of Arabic content available online, little research has been done to systematically assess and measure this content in a rigorous manner. This paper presents the results of a research project conducted by King Abdulaziz City for Science and Technology (KACST) and aimed at the development of an indicator for Arabic online content based on computational linguistic corpora. The paper is structured as follows: The next section provides a background for the research and its design, followed by a detailed description of the corpora development process and results. The paper then highlights the findings of the project and provides pointers for further research.

2 Background

The decentralized nature of the Internet's architecture makes it a very dynamic entity that remains constantly in a state of change. New hosts are added to the web on a daily basis, and many others get disconnected for several reasons. This greatly affects the accessibility to the content available on these hosts. However, the general trend has always been one of dramatic growth. The decentralization of the web extends beyond its technical architecture to encompass content production and dissemination. In today's web 2.0, any user connected to the Internet can pro-

duce, share and distribute content, using a plethora of tools and platforms for content sharing, such as Flickr, Facebook, YouTube, SlideShare, etc. A study by the International Data Corporation (IDC, 2010) reported that the world's digital output currently stands at 8,000,000 petabytes and may surpass 1.2 zettabytes this year (a zetta-byte is 10 to the power of 21).

These characteristics pose significant challenges for attempts to measure the size of content on the web. Appreciating the size of the online content is very important to understand trends in content development and growth, and in supporting the planning and implementation activities of online content initiatives. Several researchers have developed different approaches to estimate the size of the web. The common element of most of these approaches is their reliance on measuring the sizes of online search engines to infer the size of the web at large (Lawrence and Giles, 1998; Gulli and Signorini, 2005; de Kunder, 2006).

Search engines were conceived as tools to facilitate information search and retrieval on the web. The major search engines continuously crawl the Internet to index the content they find in order to make it easily accessible for their users. These indices are then exploited to provide users with the ability to search through these engines and find content relevant to their search criteria.

Given the large, and increasingly growing, number of users on the web, it is difficult to imagine that a single search engine would be capable of indexing all the content available on the web. Therefore, any single search engine would only be able to provide its users with a subset of the information available on the web (Bharat and Broder, 1998). However, due to the dramatic technological developments in the area of information indexing, search and retrieval, the major search engines can serve as an effective tool to connect users searching for information and information relevant to their search criteria. The subset of the Internet that is actively indexed by search engines is typically referred to as “the indexed web” or “the surface web” (He, Patel et al., 2007). Internet usage trends demonstrate that users rely on search engines to find information on the web. Content that is not indexed by search engines is effectively “hidden away” from normal Internet users (Capra Iii and Pérez-Quñones, 2005). Hence, it can be assumed that a reasonable estimate of the content available on

the indexed web will give a good picture of the overall online content available for Internet users.

In their study, Lawrence and Giles (Lawrence and Giles, 1998) sent a list of 575 search queries to six search engines, and analyzed the results to determine the overlap between these engines and estimate the size of the indexed web. Gulli and Signorini (Gulli and Signorini, 2005) adopted a similar approach in which they sent 438,141 one term queries to four major search engines. They then analyzed the results to estimate the size of the indexed web based on the relative size estimation of each search engine and the absolute size reported by the search engine.

De Kunder (de Kunder, 2006) argued that these approaches have inherent limitations resulting from their dependence on search engine results and overlap. He proposed a different approach that utilizes linguistic corpora in the estimation of the size of the indexed web. De Kunder implemented his approach in a project that estimated the size of the English and Dutch web.

In this research, we follow a linguistic corpora-based approach similar to de Kunder's, and apply it to the development of an indicator for Arabic online content. A critical dependency for this approach is the availability of relevant Arabic corpora that can be utilized in calculating the indicator. Due to the severe lack of Arabic computer linguistic resources in general, and those specific to online content in particular, we developed two corpora from Wikipedia and the web. The following section describes the adopted methodology and the research process in greater detail.

3 Methodology

To ensure that the estimate of the Arabic online content is reasonable and reflects the actual reality, its development should be based on a sound theoretical foundation. This foundation can be established through the concepts and ideas of computational linguistics. The most relevant of these is Zipf's law (Li, 1992), which states that in a large enough language corpus the frequency of any word is inversely proportional to its rank in the frequency table. To explain this law, the concept of word frequency must be first illustrated. In a language corpus, the frequency in which a specific word occurs in this corpus is

referred to as the word frequency of this word. A frequency table, also referred to as a frequency list, of a particular corpus is a simple table that contains each word along with its corresponding frequency in the corpus.

Based on this concept, if the word frequency of a specific word can be determined in a corpus of a known size, and in another content repository of unknown size, the size of the repository can be deduced by calculating the proportion of the word frequency in each set. Because search engines typically return the number of documents that result from a specific search query, this number can be exploited to estimate the actual size of the search engine's index. This number, however, does not refer to the word frequency in the search results, but rather indicates the total number of documents that contain the search word.

For example, searching for the word “the” on Google returns 23,990,000,000 results (“the” is the word with the highest word frequency in the English language). However, the actual word frequency for “the” will most likely be much higher than this figure, because the word will most probably appear more than once in each web page returned in the search result. To address this issue, a new concept can be introduced, referred to as “document frequency”. Document frequency indicates the number of documents in the corpus that include the word being considered, regardless of the number of occurrences of the word in each document. Following this logic, the index size of the search engine Google can be deduced based on the document frequencies in an appropriate corpus. Figure 1 illustrates this approach.

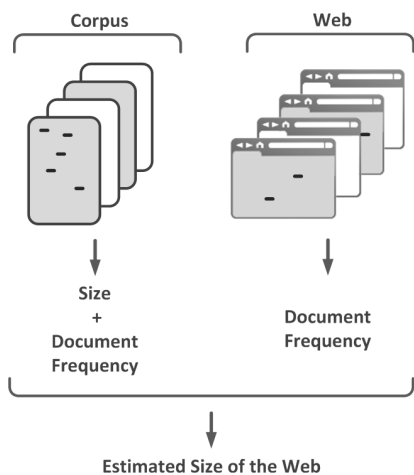


Figure 1. Estimating the Size of the Indexed Web

3.1 Corpora Building

The successful development of the content indicator depends on the availability of high quality Arabic language corpora. The corpora constitute the critical foundation that will provide word frequency statistics for the calculation of the indicator. Therefore, the first step in the development process entailed building a set of Arabic language corpora relevant to the content typical of the web.

To increase the accuracy and relevance of the indicator, more than one corpus will be used in its estimation. This will diversify the content base for the linguistic statistics and increase the corpora representation of the actual web. Three corpora were built as part of this project, using materials from the following sources:

- The Arabic Wikipedia
- The Open Directory Project
- The Arabic Contemporary Corpus

The choice of these corpora intends to diversify the fingerprints used in the calculation of the index. For example, the corpus of Wikipedia articles will have more words than the Open Directory Project because the articles on Wikipedia are mostly self-contained, while on the web pages indexed by the ODP, the same piece of content may spread over several pages. By incorporating these differences in the calculation method, more robust estimates can be made.

The corpora building process also includes the analysis and identification of the word and document frequencies in each corpus. A web-based application was developed to handle this task which will generate a table of words and their word and document frequencies in a database format. This frequency table forms the basis for the calculation of the Arabic content indicator.

Particular care was taken in the construction of the corpora from web content in order to ensure the quality of the collected materials. For example, while some sites might be targeting the Arab region, or be actually based in an Arab country, their content may not be necessarily presented in Arabic. The corpus building process will disqua-

lify any website not written in Arabic and exclude it from the corpus. Another consideration is the accessibility of the websites. Because search engines do sometimes retain links to inaccessible websites in their indices, these websites must be excluded from the corpus. This was achieved by incorporating a timeout limit on the web page retrieval in the application. Additionally, web pages that contain a very small number of words (less than 10 words) were not considered. These pages are mostly navigation panels or website introductions, and including them in the corpus will skew its linguistic statistics.

3.1.1 The Web Corpus

This corpus was intended to capture content from the public Arabic web. There are very little similar resources already available, and therefore a decision was taken to construct this corpus by an extensive crawling process on the Arabic web. To start the process, an initial list of Arabic language websites was required. The Open Directory Project (ODP) is the “largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors” (ODP, 2011). All the data available in the ODP is available under an open license that permits reuse and adaptation.

The complete directory of the ODP was downloaded and analyzed to identify the Arabic language links. These links were then extracted from the downloaded XML file and inserted into a MySQL database table. Upon completion, the “Directory” table contained information about **7,595** Arabic web sites, and was ready for use in the crawling process. The crawling process then followed the links in each of these sites and extracted their content, removed any markup, converted the content’s encoding into Unicode and stored it in a database.

After the completion of the crawling process, all pages that contain less than 10 words were removed from the database. Pages that contain such a small number of words are usually redirects to other pages or of the type similar to “Under Construction”, and would provide little value for the corpus.

By the end of the process, **75,560** pages were added to the corpus, with a total size of 530.1

MB. This constitutes the actual size of the corpus.

In order to utilize the web corpus built by the crawling process, word frequency statistics should be calculated for use in the estimation of the indicator. This process started by extracting the word occurrences in the page content in the corpus, and creating a table for these words. This exercise resulted in **659,756** unique words being added to the word list table. Word and document frequency for each of these words were then calculated by specifically designed scripts.

3.1.2 The Wikipedia Corpus

The second corpus was extracted from the Arabic Wikipedia (<http://ar.wikipedia.org>). To build this corpus, a complete dump of the database of the Arabic Wikipedia was downloaded from the official Wikipedia download site (<http://download.wikimedia.org/arwiki/latest/>) on the 24th of July 2010.

All articles that contained less than 10 words were deleted from the database, resulting in a total corpus size of **95,140** articles and 213.3 MB.

The same process described in the previous section was followed to generate the word list, document frequency and the word frequency tables for the Wikipedia corpus, using the same scripts. The extracted word list contained **760,690** distinct words, and the document and word frequencies were calculated for each.

3.1.3 Corpus of Contemporary Arabic

The third corpus was the Corpus of Contemporary Arabic (CCA), developed by Dr Latifa Al-Sulaiti at the University of Leeds (Al-Sulaiti, 2010). The content of this corpus was obtained from Arabic web sites, and although its size is rather small, it is useful to include so that the coverage of the proposed indicator is diversified even further.

The corpus was downloaded on the 25th of July 2010 from the author’s website: <http://www.comp.leeds.ac.uk/eric/latifa/research.htm>. The contents of the corpus, totaling **377** articles, were imported into a database table, and the same process for extracting the word list was

followed. The word list for the CCA contained **82,878** words. The document and word frequencies for each of these words were calculated and added to the appropriate database tables.

3.2 Selection of Search Engines

To further improve the reliability of the Arabic content indicator, and increase its representation of the indexed web, two search engines were utilized in the calculations rather than one. The selection of these search engines depends on two factors: their popularity among Arab Internet users, and the size of their search index.

Based on these criteria, the selected search engines for use in the calculation of the Arabic content indicator are Google and Yahoo!. According to ComScore, these two search engines command 65% and 20% (respectively) (comScore, 2009) of the global search engine market. Microsoft's recently launched search engine, Bing, was also considered, but the numbers of the results it returned indicate a very small index size for Arabic content, and hence it was excluded from the selection. The situation of the Arabic search engine Ayna was very similar to that of Bing. The small index size of these two search engines makes their impact on the estimated size of the indexed Arabic web negligible, and hence they were not considered.

It is crucial when more than one search engines are used to estimate the size of the indexed web that the overlap between these search engines is identified and accounted for. Naturally, the two search engines would have crawled and indexed the same pages. For example, the homepage of Wikipedia – www.wikipedia.org – can be found in the indices of both Google and Yahoo!. At the same time, the different search engines will have indexed different websites due to the differences in their crawling and information retrieval mechanisms. This is evident in the different number of results returned by the two search engines for the same search term.

To determine the overlap between the two search engines, a sample of URLs was generated by applying a logarithmic selection to the three corpora to extract 200 words from each corpus. The logarithmic selection was chosen to provide the widest coverage of the corpora according to Zipf's law. Each of these words was then sent to both search engines, and the first 10 result URLs were collected. These results were then com-

pared to determine the overlap between the two search engines.

3.3 Application Development

The Arabic online content indicator will be calculated and made available on the web on a daily basis. A dedicated application was developed to perform these calculations and present the results in a web-friendly format, including a textual and graphical representation.

The application uses a logarithmic Zipf word selection process to obtain a representative selection of words from the three corpora that were built. The logarithmic selection ensures that the selected words are representative of the actual distribution of the corpora, as it ensures that the selection covers words of widely varying document frequencies. To compile this list, the application starts with a normal series (1,2,3,4,..etc) and for each number in the list, it will calculate the anti-log(1.6). This will give the following distribution:

1, 2, 3, 4, 7, 10, 17, 27, 43, 69, 110, 176,.. etc

The application then selects the word that corresponds to each of these locations in the word list ranked by document frequency. For each selected word, the application will record the word, its word frequency and its document frequency. This Zipf-generated list will form the basis for the calculation of the Arabic online content indicator.

When the application is invoked, it will fetch the words from the Zipf-generated list, and send each word to both search engines. The application will then record the number of search results returned for each word. The size of the indexed Arabic web is then calculated, with the proper adjustments for the search engines' overlap and other considerations, such as dead links.

To ensure that only Arabic content is returned as a result of the search query, a language filter is applied to the search engine restricting the search language to Arabic. This is very helpful to eliminate sites that may include an Arabic word in their meta-data for example, but their actual content is presented in another language.

The estimate calculated by the application is then stored in the application's database, along with a timestamp to document the date and time of the

measurement. This information is utilized to present the indicator in a textual format, as well as a graphical form. Figure 2 shows the graphical representation of the Arabic Content Indicator.

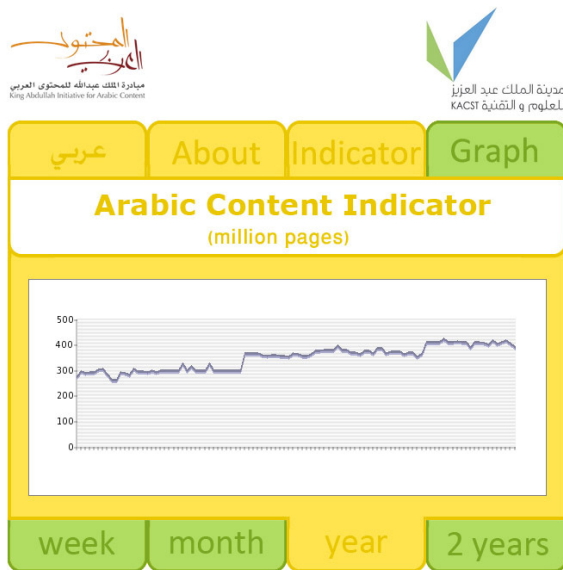


Figure 2. Arabic Content Indicator

As of the 9th of April 2011, the estimated size of the indexed Arabic web reported by the indicator exceeds 413 million pages. The application also recorded an increase of around 40% in the size of the indexed Arabic web over the six months period leading to this date. The fluctuations in the indicator's value are a natural result of the dynamic nature of the web. These happen due to changes in connectivity, hosts going offline or becoming inaccessible.

4 Conclusions and Future Work

This paper presented the design, development and implementation of a linguistic corpora-based approach for the estimation of the size of online Arabic online content. One of the major contributions of this work is the construction and analysis of three Arabic language corpora required for the development of the indicator. These corpora can be utilized by Arabic computer linguistics researchers in their studies, and constitute a significant foundation that can be further developed and built upon.

Another contribution is the development and implementation of the indicator's application that leverages the results of corpora analysis to estimate and report the size of the indexed Arabic web. The Arabic online content indicator is an important element in gauging the size of the

Arabic content on the web, and in observing and monitoring its changes over time. It is also of pivotal importance to initiatives that aim at improving Arabic content, as it enables the assessment and evaluation of the impact of these initiatives and projects, as well as informs their design and future planning.

The data that will be collected by the indicator's application is expected to have great value to researchers and practitioners working on Arabic content, and will likely lead to future research and development in this domain. Future work may include the analysis of the trends in the growth of Arabic online content and identifying correlation or causalities into factors that may affect these trends. Another interesting area might be investigating the quality of the Arabic online content to determine redundancy in such content and the topics they cover. While evaluating quality is a challenging endeavor, research into this area that leverages emerging trends of crowdsourced evaluation is very interesting and could shed light into the relevance of online content rather than focus only on quantity. The indicator can also be adapted to include geo-location data that maps the geographic source of the Arabic content and explore the comparative intensity of content production in different parts of the Arab world. The concepts and tools developed as part of this research can also be utilized in measuring the size of online content in other languages.

References

- Al-Sulaiti, L. 2010. "Corpus of Contemporary Arabic." Retrieved 25 July, 2010, from <http://www.comp.leeds.ac.uk/eric/latifa/research.htm>.
- Bharat, K. and A. Broder. 1998. "A technique for measuring the relative size and overlap of public web search engines." *Computer Networks and ISDN Systems* **30**(1-7): 379-388.
- Capra Iii, R. G. and M. A. Pérez-Quñones. 2005. "Using web search engines to find and refind information." *Computer* **38**(10): 36-42.
- comScore. 2009. "Global Search Market." Retrieved 1 February, 2011, from http://www.comscore.com/Press_Events/Press_Releases/2009/8/Global_Search_Market_Draws_More_than_100_Billion_Searches_per_Month.
- de Kunder, M. 2006. "Geschatte grootte van het geïndexeerde World Wide Web."

Gulli, A. and A. Signorini. 2005. The indexable web is more than 11.5 billion pages, ACM.

He, B., M. Patel, et al. 2007. "Accessing the deep web." Communications of the ACM **50**(5): 94-101.

IDC. 2010. "The digital universe decade - are you ready?" Retrieved 1 February 2011, from <http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm>.

Lawrence, S. and C. L. Giles. 1998. "Searching the world wide web." Science **280**(5360): 98.

Li, W. 1992. "Random texts exhibit Zipf's-law-like word frequency distribution." IEEE Transactions on Information Theory **38**(6): 1842-1845.

Mansell, R. and U. Wehn. 2000. Knowledge societies: Information technology for sustainable development, United Nations Pubns.

ODP. 2011. "The Open Directory Project." Retrieved 1 February, 2011, from <http://www.dmoz.org/docs/en/about.html>.

Stone, D. and S. Maxwell. 2005. Global knowledge networks and international development: bridges across boundaries, Routledge.

Taxonomy of personalisation for Generating personalised content in Technical Support Forums

Solomon Gizaw
University of Limerick
Localisation Research Center,
Solomon.gizaw@ul.ie

Jim Buckley
University of Limerick
Localisation Research Center,
Jim.buckley@ul.ie

Abstract

There are two traditional approaches to meeting international users' requirements during content generation. The first is localisation which adapts a content to meet the language, cultural and other requirements of a specific locale. The second is personalisation which involves delivering relevant content and presenting information according to individual users' preferences.

The fundamental question that must be answered in generating personalised content is: what are the relevant attributes for personalising content? Work to date in personalisation has been based on several logic-based standards and frameworks that have been proposed. However these different standards have led to a degree of inconsistency in the field and are open to accusations of validity.

This research aims to empirically identify the relevant attributes for personalising content. It focuses on data obtained from technical support forums on the web, a growth area in terms of customer support. It uses a grounded-theory based approach to analyse the conversations on these forums in order to identify the personalisation attributes of the support provided. This paper reports on a preliminary study in this work, which analyses data from a number of technical support forums, and presents an initial taxonomy of empirically derived personalisation issues for this domain.

1 Introduction

With the growth of the WWW (World Wide Web) and the Internet, the necessity to address the needs of global markets with different cultural and individual requirements has also increased (Capstick et al. 1998). There are two traditional approaches to meeting such user requirements. The first is localisation which adapts

a content to meet the language, cultural and other requirements of a specific target market "locale"(Microsoft Press 1999). The term "locale" in this context refers to some specific regions or country. The second is personalisation that is primarily concerned with ensuring that information can be delivered to different end users in a format which reflects their specific needs (Gibb and Matthaiakis 2007).

In this context, localisation considers culture as a collective behavior or characters of a community who is living in some specific region or country. The information generated from the localisation process is expected to represent the cultural needs of that community. However the uniqueness of individual interests is not necessarily addressed in the current localisation process (H. Sun 2004). The same author, Sun, also stated that current localisation practices suffer from a narrow and static vision of culture resulting in usability problems for IT product and design. Localisation can therefore be seen as an intermediate stage before full personalisation (F. Gibb and I. Matthaiakis 2006).

On the other hand, personalisation involves delivering relevant content and presenting information according to individual users' preferences. These preferences are gathered explicitly or implicitly from the users to build a user model. Unlike localisation, the consideration is not bounded by locale rather; it goes beyond community interest and incorporates individual preferences.

There has been a plethora of research in the area of personalisation (J. Vesanen 2007, Miceli et al 2007, P. Ralph and J. Parsons 2006, A. Tuzhilin and G. Adomavicius 2001, K. Instone 2004, A. F. Smeaton and J. Callan 2005). This research

has proposed systems for personalisation of material and proposed different sets of attributes of personalisation, upon which these systems should be built. However, these proposed attributes are all theory-based and thus are open to accusations of being inconsistent and misguided. Thus it is the aim of this research to define personalisation attributes empirically and to rank these attributes in terms of order of importance for specific domains. The insights we ultimately derive will provide an empirical foundation for performing personalisation on content and constructing user models. Hence, this work attempts to develop an empirically-derived taxonomy of personalisation, to complement and enhance the existing theory-based taxonomies.

This paper reports on a preliminary study in that context where we use aspects of Grounded theory to identify the attributes exhibited in a web-mediated technical support domain. Technical support, as defined by (Das 2003), is a post sales service provided to customers of technology products to help them incorporate a given product into their work environment. It not only serves to improve the users' needs with respect to the product but can also provide a source of income for companies that provide the support services.

2 Literature Review

Personalisation is the process where customers receive different treatments based on their past behavior. These treatments are concerned with what is good for the business, serving the customers, and improving the quality of their resulting experience (Instone 2004). Thus personalisation describes the problem of how to use customer's information to optimize a business's relationship with its customers (Sahai and Kuno 2002).

The three core motivations for personalisation, from a user's perspective are (Blom 2000): to access information effectively, to accomplish a user's work goal, and to accommodate individual differences. To accomplish these core motivations, the implementation of personalisation has three dimensions: what to personalise (content, interface, service, modality), to whom to personalise (individuals or categories of individuals) and who does the personalisation (is it implicit or explicit) (F. Haiyan and P. M. Scott 2006).

With regard to this 2nd dimension, personalisation requires user information in order to create a user profile which can be used to identify, classify, store, filter, organise and present content which matches that individual's needs (F. Gibb and I. Matthaikakis 2006). Various personalisation applications can contain different types of attributes about individual users. However, in many applications, this attributes generally can be classified into two basic types – demographic and transactional, where demographic describes who the user is and transactional describes what the user does (A. Tuzhilin and G. Adomavicius 2001).

Different standards have been defined in the literature by different standard bodies, to identify and classify personalization attributes. Table 1 shows the categories of classification of personalization attributes as defined by some of these standard bodies. These attributes in turn are used as the basis for personalisation of content by many researchers in the field. Table 2 show some examples, detailing the personalisation attributes used by different researchers.

However, this literature basically is based on what researchers and service providers think are the user personalization issues rather than what the users actually want and which attributes really matter in the process of generating personalised content. Even though there has been an amount of work in implementation of personalisation, individual researchers have based their approaches on achieving personalisation goals that, while intuitively correct, have never been empirically evaluated as the core or even the correct, personalisation goals. This work attempts to address this by empirically deriving personalization attributes of relevance to the user.

3 Motivation

Many organizations have moved their customer care technical support from the product manual to the World Wide Web. Traditionally they have adopted a one-size-fits-all approach by delivering a FAQ and simple knowledge base search engine (Steichen and Wade 2010).

No	Standard Body	Name	Purpose	Attributes
1	W3C	P3P	standard for profile security	Demographic attributes (Identity, Age, Revenue) Professional attributes (Employer, Job category, Expertise) Behaviour attributes (Trace of previous queries, Time spent at each navigation link)
2	Telematics Information Engineering Project Number 8011	User profiles	For digital libraries	Personal data (Identity) Collected data (Content, Structure and origin of accessed documents) Delivery data (Time and support of delivery) Behavioural data (Trace of user-system interactions)
3	IEEE	Learning Object Meta-Data (LOM)	Educational Purpose	Educational Difficulty (Hard, Easy) Interactive Type (Active, Mixed, Expositive) Interactive level (low, medium, high) General Life Cycle
4	Dublin Core Metadata Initiative	Dublin Core Metadata Elements	Core Metadata standards	Title Subject Description Creator Publisher

Table 1 - Personalization Standards: Bodies and attributes

Attributes	Instances	References	Related to No
Product state	Product installation state Configuration state Pro-active actions Re-active actions	Steichen and Wade 2010	3
Knowledge state	Novice Expertise Procedural Specialised	Alba and Hutchinson 1987	3
User Values	Privacy Security Trust Brand Price	Cranor et al. 2002 Bart et al. 2005 Wind and Rangaswamy 2001	2
Orientation	Goal Oriented Utilitarian Hedonic	Celsi and Olson 1988 Hoffman and Novak 1996 Wolfinbarger & Gilly, 2001	1
Behaviour	Beliefs Interest E-Joyalty Involvement	Sun 2004 Reichheld and Scherer 2000 Srinivasan et al. 2002 Hoffman & Novak, 1996 Novak, Hoffman, & Yu-Fai, 2000 Anderson and Narus 1984	2
Demographic make ups	Gender Age Income	Sun 2004 Bouzaghoub and Kostadinov 2006 Tuzhilin and Adomavicius 2001	1
Content preference	Procedures content Consult Explanations Overview first	Steichen and Wade 2010	4
Process Type	Activity : Task Concept : Narrative Text, Table, Image, Summary	Novak, Hoffman, & Dubachek, 2003	
Educational	Difficulty (Hard, Easy) Interactive Type (Active, Mixed, Expositive) Interactive level (low, medium, high)	Steichen and Wade 2010	3

Table 2:- Categories of Personalisation Attributes from the Literature

In contrast, the internet-based support systems that will thrive in the next generation will have to overcome the existing language and cultural barriers, particularly in applications operating in an international business environment (Schütz 1996). In addition, in order to meet user requirements to provide effective service for individual users, tailored to their preferences, service providers not only need to localise the content, but also to personalise it.

One problem with moving to this personalized support environment is that the research work in this area has been based on achieving personalisation goals that, while intuitively correct, have never been empirically evaluated as the core or even the correct, personalisation goals. This study will move towards addressing this gap in the literature by conducting an empirical study to identify the core attributes of personalisation in customer support.

4 The empirical Study

4.1 The Research question

The fundamental question that must be answered in generating personalised content is: (1) how can we generate content beyond localisation to satisfy user requirements? This, in turn is based on: (2) what are the relevant attributes for personalising content, and presentation of information of interest to users? These questions have not been addressed consistently in many areas, including the customer care technical support area. There is a need to have an empirical evidence of the core personalisation issues and thus to have a clear definition and measurable goal as a guide for generating personalised content.

4.2 The Design

This experiment has four phases: Selection of unit of analysis, data analysis method, coding process and interpretation and categorising process.

4.2.1 Unit of Analysis

Considering the above limitations with regard to personalised content development, the paper will attempt to empirically identify relevant personalisation issues that arise in community support forum conversations and rank these issues towards characterising and generating personalised content in a customer care scenario. Community support forums are used because research has shown (Oxton 2010, Steichen and

Wade 2010) that customers are abandoning official technical support facilities provided by companies and increasingly migrating to community forums for their technical support, suggesting that these forums are giving customers the individual care they require.

Before we continue further let's define some of the terms which are used in this paper: A thread is a question posted in the forum with its responses from the community. There can be many threads in a forum. A message is the response of a question poster or other community participants to respond for the question. There can be one or more messages in one thread. We decided to include technical support forums which are the most popular ones from the Google search ranked list since the Google search ranks according to most visited forums. We have looked at their characteristics in terms of number of responses and time-delay of responses. At last we performed aspects of grounded theory on the individual messages in threads in these community forums.

The empirical first phase analysis is conducted on selected 7 IT technical support forums looking at a total of 31 threads which are categorised as: General Hardware, Networking, Security and virus, and Windows as shown in the figure 1.

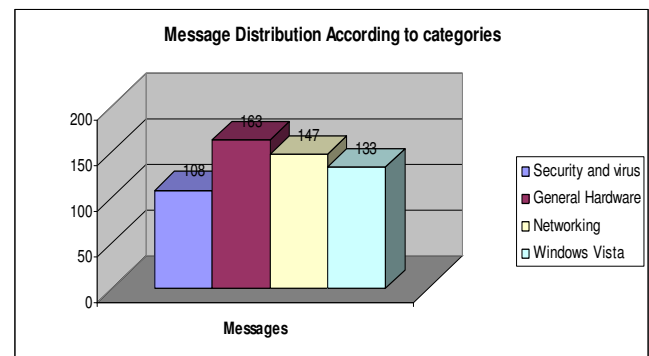


Figure 1: Message distribution According to Categories

The messages between the community forum member in each forum ranges between 3 and 54 with a total of 551 messages. The average messages per each thread are 17.8. The response time for each technical support request post ranges between 5 minutes to 5 Days. However 65% of the posts have responses within one or two hours.

4.2.2 Data analysis method

This study incorporates issues including the users' emotional and social drives and perspectives, their motivations, expectations, trust, identity, social and cultural norms. Consequently, we will use an inductive approach, driven from rich data, and employ the appropriate qualitative empirical analysis.

In these forums, we intend to employ statements as our data analysis unit of choice. A member can raise a question and post it so that other forum members can read and respond. Statements will initially be analysed for instances when users' signal that their needs are not addressed by the current response, as such statements strongly signal that an individual information need is not being met. The collected data set will be analysed to identify users' interests. One way of doing so is using the Emergent/Open Coding approach. The emergent coding approach is one of the techniques used to analyse text content in a qualitative research. It first examines a subset of the data independently and each data set develops a list of key coding categories based on interpretation of the data (Lazar et al 2010).

This research scenario also demands a non-predefined coding schema, so the method of Open Coding is suitable in this context. Open coding is the part of the analysis concerned with identifying, categorizing and describing phenomena found in the text. It analyses and identifies any interesting phenomena in the data (Lazar et al 2010). Each derived phenomenon will be coded. At last, similar instances are combined together in new ways after open coding by making connections using Axial Coding.

5 Results

5.1 Characteristics of community forums

After selecting the community forums for analysis, the coding of the messages is performed. This is done using open coding and in vivo coding to sort concepts into categories, so that in each category the concepts are both as similar as possible to each other, and as different as possible from concepts in every other category identified. Table 3 shows the characteristics of the forums categorised and their instances, examples and number of counts for each category.

6 Conclusion

Generally the number of responses on each thread on average and the time of responses for each post show the community forums are making efforts to deliver relevant and personalised information for their users and this shows that it is the right place to find characteristics and behaviors of different users.

A preliminary grounded finding of the messages shows users are primarily concerned with experience, trust, user-values, emotions and constraints respectively. User experience is main important issue to consider delivering personalised information. Characteristics of Emotion are OK, but it sometimes needs moderation. This shows that, the level of novice and expertise must be defined and categorised according to domain specific. Personalisation issues must have a system to identify the users' domain specific knowledge related to his questions. The taxonomy of personalisation already present doesn't put the priority of personalisation attributes into consideration.

The intention of responses intended to deliver only the relevant knowledge which doesn't consider other personalisation attributes. Even if there is a lot of discussion that are performed sometimes it ends without any solution for the user because of many misleading speculation and suggestions. However, sometimes good community forum participants try to understand the situation and try to answer accordingly in a way that the user can understand and use the information.

In the future much more analysis needs to be done with wider directed samples for more theory building and saturation.

Acknowledgments

This research is based upon works supported by Science Foundation Ireland (Grant Number: 07/CE/I1142) as part of the Centre for Next Generation Localization (www.cngl.ie).

Category	Instances	Example	Number of occurrence	Coding Type
Experience	Novice	<ul style="list-style-type: none"> ➤ I'm a total novice ➤ have little knowledge ➤ have no clue 	22	INVIVO
	Expertise	<ul style="list-style-type: none"> ➤ Im an IT Technician ➤ Ive got a pretty good idea what im doing 	14	Open Coding
Trust	Distrust	<ul style="list-style-type: none"> ➤ I'm not about to put my credit card info ➤ I'm afraid to delete it 	10	Open Coding
	Degree of Trust	<ul style="list-style-type: none"> ➤ Tend to agree ➤ You are probably tell the truth 	6	Open Coding
User Values	Price	<ul style="list-style-type: none"> ➤ Cheaper Price ➤ No cost ➤ No need to pay 	10	IN VIVO
	Brand	<ul style="list-style-type: none"> ➤ Blame Your vendor ➤ Your vendors don't want to support 	5	Open Coding
Emotional	Anger	<ul style="list-style-type: none"> ➤ It's hard to soar like an Eagle when you are flying with Turkeys ➤ Would you cut off your legs while running a race? ➤ You don't know what you are talking about ➤ 	11	Open Coding
	Emphasis	<ul style="list-style-type: none"> ➤ really handy thing called SEARCH ➤ QUITE helpful ➤ Very important ➤ Disabled your Firewall 	8	Open Coding
	Frustration	<ul style="list-style-type: none"> ➤ It's hard to soar like an Eagle when you are flying with Turkeys ➤ Would you cut off your legs while running a race? ➤ You don't know what you are talking about 	4	Open Coding
	Stress	<ul style="list-style-type: none"> ➤ Oops sorry for the wrong doing, Shows how stressed i was 	3	Open Coding
Constraints	Moderate User	<ul style="list-style-type: none"> ➤ I am a volunteer here with a job and family so I ask that you be patient when waiting for replies. 	9	Open Coding
	Moderate Answer	<ul style="list-style-type: none"> ➤ Just FYI, first one must ascertain what is wrong before attempting to fix the problem. Just trying different fixes willy-nilly in hopes of resolving the problem is a waste of time and energy and more likely to make things worse than better. 	4	Open Coding

Table 2:- Categories of Personalisation Attributes from the Literature

References

- Adomavicius G. and Tuzhilin A. 1999. *User Profiling in Personalization Applications through Rule Discovery and Validation*, KDD-99
- Alba J. W. and Hutchinson J. W. 1987. *Dimensions of Consumer Expertise*, Journal of Consumer Research
- Blom J. 2000. *Personalization: a taxonomy*, extended abstracts on Human factors in computing systems, The Hague
- Bouzeghoub M. and Kostadinov D. 2006. *Data personalization: a taxonomy of user profiles knowledge and a profile management tool*, Technical report ACI APMD, project MD-33
- Capstick J., Diagne A. K., Erbach G., Uszkoreit H., Cagno F., Gadaleta G., Hernandez J. A., Korte R., Leisenberg A., Leisenberg M. & Christ O. 1998. *MULINEX: Multilingual web search and navigation*, In Proceedings of Natural Language Processing and Industrial Applications
- Celsi R. L. and Olson J. C. 1998. *The Role of Involvement in Attention and Comprehension Processes*, Journal of Consumer Research
- Cranor L., Langheinrich M. and Marchiori M. 2002. *A P3P Preference Exchange Language 1.0*, W3C Working Draft <http://www.w3.org/TR/P3P-preferences/>
- Das A. 2003. *Knowledge and Productivity in Technical Support Work*, Management Science
- Gibb F. and Matthaikakis I. 2007. *A framework for assessing web site localisation*, Electronic Library, Vol. 25 Iss: 6, pp.664 – 678
- Hoffman D. L. and Novak T. P. 1996. *Marketing in Hypermedia Computer-Mediated Environments: Conceptual Foundations*, Journal of Marketing
- Instone K. 2004. *An Information Architecture Perspective on Personalization*, Designing Personalized User Experiences in eCommerce, pp. 75-93
- Kirsh D. 2000. *A Few Thoughts on Cognitive Overload*, Intellectica, 30(1):19–51
- Lazar J., Feng J. H. and Hochhesier H. 2010. *Research Method in Human–Computer Interaction*, John Wiley & Sons Ltd publisher
- Microsoft Press 1999. *Computer Dictionary*
- Oxton G. 2010. *The power and value of on-line communities*, Consortium for Service Innovation in keynote, April CNGL meeting
- Reichheld F. F. and Schefter P. 2000. *E-loyalty*, Harvard Business Review, 78, pp 105–113
- Sahai A. and Kuno H. 2002. *My Agent Wants to Talk to Your Service: Personalizing Web Services through Agents*, in proceedings of the first international workshop on challenges in open agent systems, pp. 25--31.
- Schütz J. 1996. *Combining Language Technology and Web: Technology to Streamline an Automotive Hotline Support Service*, In: Proceedings of AMTA-96, Montreal, Canada.
- Sun H. 2004. *Expanding the Scope of Localisation: A Cultural Usability Perspective on Mobile Text Messaging Use in American and Chinese Contexts*, Rensselaer Polytechnic Institute Troy, New York
- Steichen B. and Wade V. 2010. *Adaptive Retrieval and Composition of Socio-Semantic Content for Personalised Customer Care*, Centre for Next Generation Localisation
- Wind J. and Rangaswamy A. 2001. *Customerization: The Next Revolution in Mass Customization*, Journal of Interactive Marketing, 15(1), pp13–32

Content independent open-source language teaching framework

Randil Pushpananda

Language Tech. Research Laboratory
University of Colombo
School of Computing, Sri Lanka.
rpn@ucsc.cmb.ac.lk

Chamila Liyanage

Language Tech. Research Laboratory
University of Colombo
School of Computing, Sri Lanka.
cml@ucsc.cmb.ac.lk

Namal Udalamatta

Language Tech. Research Laboratory
University of Colombo
School of Computing, Sri Lanka.
ugn@ucsc.cmb.ac.lk

Ruvan Weerasinghe

Language Tech. Research Laboratory
University of Colombo
School of Computing, Sri Lanka.
arw@ucsc.cmb.ac.lk

Abstract

Language is a very powerful tool for mutual understanding between people. In the increasingly globalized society we live in, there is great importance and interest in learning languages other than one's own mother tongue. Computer based teaching tools provide a potent medium through which such demand can be met conveniently. *Shikshaka*, a computer based teaching framework was developed for teaching languages using a dialogue-based andragogy and is aimed to fulfill the above requirement. Effective technological methodologies were used to develop the system as an interactive tool using the goal oriented approach developed by linguistic scholars. We have used the framework to implement two scenarios: to teach Tamil using Sinhala and to teach Sinhala using English. The learner's language can be customized with little effort, while the framework is flexible enough to be customizable to teach other target languages as well with some pedagogical input.

1 Introduction

Listening, speaking, reading and writing are four main aspects of learning a new language. Due to the increased interest in learning new languages, language learners are demanding more and more user-friendly, less disruptive resources to fulfill this need. Globalization is the main reason for

the increasing interest in learning second, third and foreign languages. Ease of travel, advances in technology, and internationally focused economic systems are the benefits which can be obtained through language learning. In addition to the above, there are other motivating factors to learn new languages such as understanding alien cultures, increasing the number of people on the globe with whom you can communicate, and to make travel more feasible and enjoyable, among others (Rane and Sasikumar, 2007). The lack of knowledge in a second language on the other hand has been the cause of many misunderstandings and even causes for feuds and wars. Sri Lanka's ethnic conflict can be argued to have its roots in language among other factors. No local language project could ignore the strategic opportunity provided by technology to scale the teaching and learning of another language.

Learning a new language indeed is not an easy task for most humans. It requires proper guidance and materials. There are various kinds of materials available for teaching and learning languages, each using their own pedagogical and methodological approaches to teach language. Rane and Sasikumar (2007) showed that, learning new languages using books, magazines and courses conducted by the native language experts are traditional approaches, and that these traditional approaches do not meet some of the demands of modern life, which require *anytime* learning and *on-demand* learning.

Some of these requirements can be addressed effectively by developing computer based learn-

ing tools that can assist people in learning new languages. Effective use of such technology has the potential to overcome some of the limitations of traditional language learning contexts.

In Sri Lanka, effective use of computer technologies in the language learning process have not been reported in the literature. Primarily this is due to the fact that not much research work has been carried out in this area. Sri Lanka is a country with a multi-ethnic, multilingual and multi-script environment. In such an environment, learning several languages is advantageous and essential for the entire population. Considering these and other facts, we mainly focused our research on developing an open-source and customizable language teaching-learning framework to develop a Computer Assisted Language Learning (CALL) tool to teach any (target) language using any other (source) language. By using such a framework and as a proof of concept, we targeted to develop two language learning scenarios: to teach Tamil in Sinhala and Sinhala in English.

The rest of this paper is organized as follows. Section 2 summarizes the related work in this area; Section 3 describes the language teaching and learning concepts; Section 4 describes the methodology used to build the language teaching framework; Section 5 describes the design and implementation of the system; Section 6 describes the tools were developed using the teaching framework. Finally the paper concludes with a summary of the current status and looking future works.

2 Related Work

There is not much literature available on the use of CALL tools in teaching non-Latin languages. Most learning and teaching tools available are for teaching English as a second language using proprietary, closed systems.

Vsoft is an educational software development company in Sri Lanka which produces CALL software for the use of students who are preparing for various exams.

In addition to such CD-ROM based tools, there are some web based tools for teaching Sinhala such as *Let's Speak Sinhala*, *Learn Sinhala* (Nick and Nishanthi, 2008; Vsoft). These are targeted at helping adults and children (respectively) learn the basics of spoken Sinhala.

CALL tools for other languages such as Indic languages are available to some extent. *Marathi Tutor* is a web-based constructive learning envi-

ronment for teaching spoken or conversational Marathi. Rane and Sasikumar (2007) showed that the above framework covers basic vocabulary and construction of simple and compound sentences, thus enabling the learner to converse fairly well in common places such as at a bank or post office.

MarathiMitra (MarathiMitra) is also a commercial web based tutoring system for teaching spoken Marathi. In this system, the medium of instruction (source language) is English. This is neither an open-source nor a content independent framework.

RosettaStone (Rosetta) is also a commercial computer based language learning software, which supports some 20 languages using rich visual imagery to help students learn and think in a new language. Even though this tool supports Indic language such as Hindi, it still does not support Sinhala or Tamil.

The University of Bologna in Italy has developed a multimedia framework for second language teaching in self-access environments under the *DIAPASON* project. Tamburini (1999) showed that *DIAPASON* enables the teaching of English to university students up to an intermediate level, by building a self-access environment freely available to learners.

Many existing commercial and proprietary language learning tools do not support Unicode, are sometimes complex for non-technical learners to handle, and most importantly are not based on language teaching pedagogy. In addition, most of the above tools are hard-wired for teaching a particular language, and do not provide a framework for linguists to develop effective language learning environments.

3 Second Language Teaching and Learning

Second language teaching is a research area of Applied Linguistics. When someone learns a language as native in the speech community is called the language acquisition. Learn a new language other than the native we called second language or foreign language learning. These two terms can be misleading and it makes sense by their definitions. For example, Tamil and English languages are second languages for a native Sinhala speaker while Chinese and Japanese are foreign languages for them. Similarly languages like Sinhala and English are foreign languages for Japanese.

In applied linguistics, a broad range of research activities on second language teaching has been carried out. Pedagogical methodologies which have been identified in language teaching and learning were used in developing the *Shikshaka* framework. Contents developed for the framework are based on the study of second language learning pedagogy.

4 Methodology

4.1 Pedagogical consideration

The methodology we used to develop this framework can be divided into two steps. As the first step we studied the spoken language teaching methodology by examining the courses which were developed as textbooks (Gair et al., 2005; Karunatilaka, 2004; Fairbanks, 1968; Gunasekara, 2008) and computer based educational systems (RosettaStone, Nick and Nishanthi, 2008). It was clearly identified that the most effective way of teaching spoken language is through conversations at general places and situations such as at restaurants, post offices, police stations in order to facilitate the more effective *situated* kind of learning in adults. This situation based language learning methodology has been shown to lead to effective grasping of language constructs compared to formal abstract teaching. As the second step we studied how to increase the interactivity and attractiveness of the language tools and how to implement functionality for this within the framework. It was clearly identified that the use of real still images, callouts and audio files will increase the above two factors. Callouts were used to display the phonetic representation of utterances and are aimed at helping users with pronouncing words of the target language.

The framework has been designed in such a way that language teachers can add contents to the grammar section of the framework according to their preferences. In addition to the above, three types of exercises were introduced to practice the learning which gain through the lessons. It includes two word matching exercises and one sound matching exercise. Sound matching exercises are used to help learners identify words and pronunciation, while word matching exercises are used to help them identify different word classes.

4.2 Technical consideration

Our research identified that most of the frameworks and tools have some deficiencies such as

embedded content (i.e. content is hard-wired with the framework and they cannot be separated), non localizability, platform dependency, lack of Unicode support and difficulties of adding and modifying content.

Our research proposed a solution to address the above deficiencies and provide users with multimedia rich interactive language teaching environment. This is primarily achieved by separating the course content from the framework. XML technology was used to keep the course content independent so that the tool can be easily adapted and localized for the other languages and cultures. The framework also facilitates the changing of the sequence of lessons as required.

5 Design and Implementation

The content management framework and the user interfaces of the system are implemented using *Adobe Flash* technology. Flash provides state-of-the-art rich Internet application technology for developing interactive media rich tools over the web. *ActionScript* was used for communication between user interface and XML files in developing the on-demand loaded content for the flash application. The choice of Flash enables the delivery of an efficient, lightweight, platform independent, multimedia rich and web-enabled system. Proposed combined architecture which combines both Flash objects and XML files provides a flexible way of adding and deploying fully customizable content (lessons) rapidly. One important gain in using this combination of technologies was that it was able to overcome the issue of flash's lack of support for Sinhala Unicode rendering. This benefit accrue to any new language which involves complex scripts which are currently not supported by flash.

Since XML stores the information in a more accurate, flexible and adaptable way, the XML technology was used to hold data for the framework and maintain the structure of the data. The use of XML makes the content of the framework more readable and more extendable. Each type of course contents is organized in one or more XML files. The framework is engineered in a manner so that it can be readily customized for teaching other subjects in other languages.

Each of the lessons which include dialogues, the grammar and exercises can be separated into chapters from 1 to n . The *Shikshaka* framework was developed to support any number of sections (referred to as chapters) and is flexible enough to extend itself to be used by any two languages.

Each chapter (i.e. chapter folder) contains dialogue, grammar and exercise folders (Figure 1) and each folder type has its own file structure. Separate XML files are maintained inside each folder to reduce the complexity of the framework.

5.1 Dialogue

The framework is designed to use images, voices, text and callouts to make the dialogues more attractive (Figure 2). Users can customize these features according to their requirements.

Images: Since Human expressions can be shown clearly by using digital photographs we used digital images for conversations. The framework supports for still images with 1024x768 resolution and most of image file formats such as .JPG, .PNG and etc.

Voices: Exact pronunciations of words of both target language and source language are given by pre-recorded voice clips. Popular audio formats like .mp3, .wav and etc. can be used for voice clips. Name of the file, wave format and file path of voice clips can be changed by editing the XML file.

Text: Three types of texts are shown namely; text in target language script, text in source language script and the pronunciation of target language's words in transliterated form. All these features can be customized as user preferred.

Callouts: Callouts are used to show the text which mentioned above in an attractive way to the users. By editing the XML file, callout positions (x and y coordinates) and size of the callout can be changed.

XML file (dialogue): We designed an XML file for conversations (Figure 3) and maintained XML files for each and every conversation separately. It includes source language text, transliterated text, target language text, font sizes, font colors, font names, coordinates of the image file, size and name of the image, names of the source and target audio files, name of the callout and etc.

5.2 Grammar

The grammar being used in the dialogues can be elaborated in the grammar section by adding the relevant grammar in to the framework. The framework was designed to add titles and descriptions related to the title as two separate XML files and a separate cascading style sheet (CSS) was designed to handle the formatting of contents inside the XML data file.

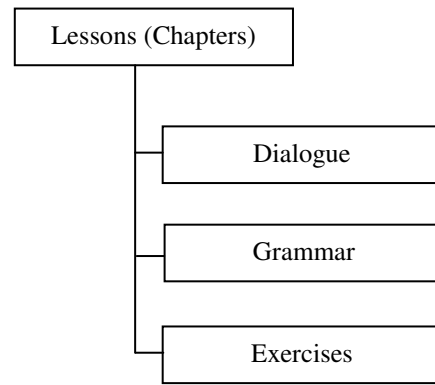


Figure 1. Main folder structure of the framework.

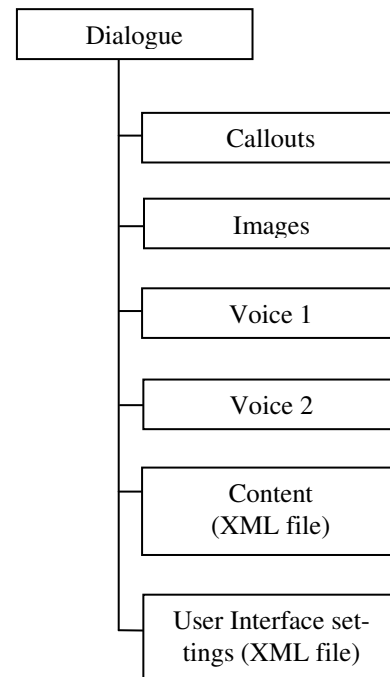


Figure 2. Structure of a dialogue.

```

<person name="பித்தர்">
  
  <audioclip lang="ta" src="vimal_1.mp3"/>
  <audioclip lang="si" src="vimal_s_1.mp3"/>

  <callout>
    
    <text lang="ta" x="585" y="228" height="100" width="200"
      fontname="Malithi Web" fontsize="35" fontcolor="0x0000FF">
      என்ன இருக்கிறது?</text>
    <text lang="tr" x="585" y="258" height="100" width="200"
      fontname="Malithi Web" fontsize="30" fontcolor="0x000000">
      එන්න ඉරික්ක ?</text>
    <text lang="si" x="585" y="288" height="100" width="200"
      fontname="Malithi Web" fontsize="30" fontcolor="0xFF0000">
      මොනවාද නිවෙන්න?</text>
  </callout>
</person>
  
```

Figure 3. XML structure for dialogue.

5.3 Exercises

The framework has designed to include relevant exercises of a particular chapter into the exercise section. Contents of the exercises are kept in separate XML files and three mapping exercises were designed as mentioned in the methodology section. To increase the attractiveness of the exercises, animated gif images were used to indicate correct and wrong answers.

6 Experiments

Two CALL scenarios were developed using the *Shikshaka* framework to teach Sinhala through English and Tamil through Sinhala. Figure 4 and Figure 5 show screenshots of the front page of the two case studies (refer Appendix A for examples of the user interface design of these courses). Developers or language teachers can use the *Shikshaka* framework to design the language learning tools conveniently for any target language using a source language.

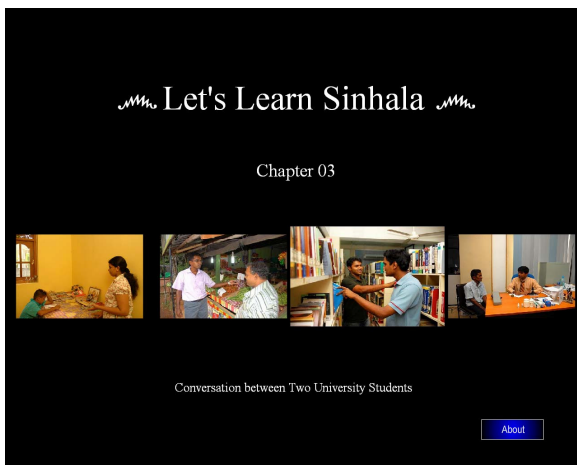


Figure 4. Sinhala learning tool.



Figure 5. Tamil learning tool.

These two scenarios were released under Creative Common public licensing (attribution, non-commercial and share alike) category. Samples of both tools were distributed freely in CD format especially in exhibitions and workshops to get feedback from users. School teachers have recommended distributing these tools through the school network as students can be motivated with IT enabled teaching methods. In addition to the above, both courses are publicly available at <http://www.e-learning.lk/vle/course/>. The courses are meant to be facilitated by a teacher, but may also be used for self-learning by the disciplined user. The full content together with the source code and documentation will also be available for download from the www.ucsc.lk/trl

7 Conclusions and Future Work

The *Shikshaka* framework was developed to address the problems identified in section 4.2 of most teaching tools being proprietary and hard-wiring content with framework. Two language learning courses were developed for teaching Tamil through Sinhala and Sinhala through English as a proof of concept for the framework.

Plans are underway to distribute the two courses in CD/DVD format in addition to their access through the web. Controlled offering of the courses are also planned in order to solicit feedback in order to evaluate the effectiveness of the framework and pedagogy. In addition to the above there is some demand to include an evaluation system to the framework for each lesson as well as revision exercises after each five chapters.

Moreover, it is expected to provide a more convenient content entry method for the framework without the need for editing coordinates and sizes of the XML data. This will help to improve the user-friendliness of the framework. Finally the *Shikshaka* framework will also be extended to support levels of study.

Acknowledgment

This work was carried out under PAN localization project phase 2 funded by IDRC Canada. We are immensely grateful to authors of the book; "*An Introduction to Spoken Tamil*" W. S. Karnatilaka, James W. Gair, and S. Suseendira-jah and the author of the translated version of the above book W. M. Wijeratne for providing us the contents for developing the Tamil learning tool. The contribution made by Tissa Jayawardena for reviewing the contents of the Sinhala learning

tool is also acknowledged. We would also like to thank our colleagues at the Language Technology Research Laboratory, UCSC who provided insight and expertise that greatly assisted this research, including those who have left us since then. We also thank all the staff at University of Colombo School of Computing (UCSC) for giving us great support to develop the two courses. Authors would also like to acknowledge the feedback given by two unknown reviewers to improve the quality of the work reported. Any remaining errors are our own.

References

- G. H. Fairbanks, J. W. Gair, M. W. S. De Silva. 1968. *COLLOQUIAL SINHALESE*. Cornell University.
- J. W. Gair, S. Suseendirajah, W. S. Karunatilaka. 2005. *An introduction to spoken Tamil*. Godage International publishers, Sri Lanka.
- A. M. Gunasekara. 2008. *A comprehensive grammar of the Sinhalese language*. Godage International publishers, Sri Lanka.
- W. S. Karunatilaka. 2004. *An introduction to spoken Sinhala*. M. D. Gunasena and Co. Ltd, Sri Lanka.
- MarathiMitra .
www.marathimitra.com
- Nick and Nishanthi. 2008. *Let's Speak Sinhala*.
http://www.speaksinhala.com/
- A. Rane and M. Sasikumar. 2007. *A Constructive Learning Framework for Language Tutoring*. In: Iskander, M. Innovations in e-learning, instruction technology, assessment, and engineering education. Dordrecht Netherlands: Springer.
- Rosetta.
www.rosettastone.com
- F. Tamburini. 1999. *A multimedia framework for second language teaching in self-access environment*. Computers & Education 32 (2): 137-149.
- Vsoft.
http://vsoft.lk/

Appendix A. Design of the tool

The Sinhala learning course consists of 15 chapters while Tamil learning tool consists of 25 chapters. Each chapter contains a dialogue, a grammar and an exercise. Figures 6, 7 and 8 show the designed user interface for dialogue, grammar and exercises respectively.



Figure 6. User interface – Dialogue

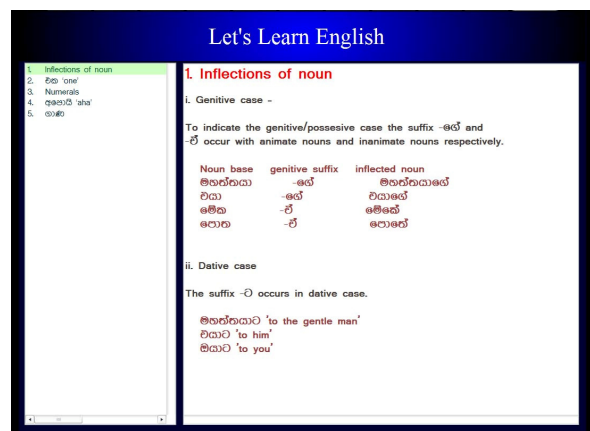


Figure 7. User interface – Grammar

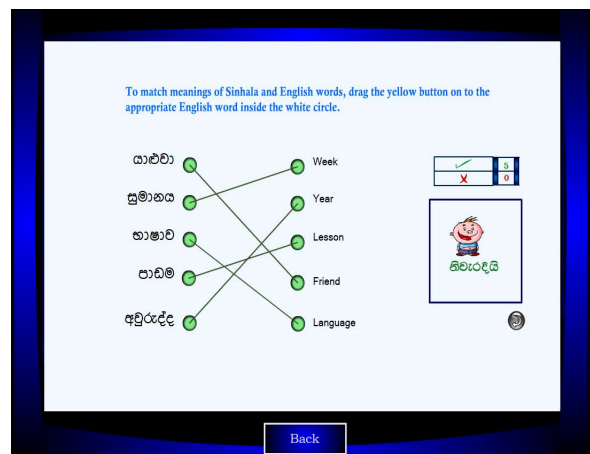


Figure 8. User interface - Exercise

English to Sinhala Machine Translation: Towards Better information access for Sri Lankans

Jeevanthi Liyanapathirana

University of Colombo
School of Computing
Sri Lanka

juliyapanapathirana@gmail.com

Ruvan Weerasinghe

University of Colombo
School of Computing
Sri Lanka

arw@ucsc.cmb.ac.lk

Abstract

Statistical Machine Translation is a well established data driven approach for automatic translation between different languages. However considerably few researches have taken place on Sri Lanka as well as Asian Languages. We research on possible English-Sinhala Translation using Statistical Machine Translation. Special attention is paid for the impact of parameter tuning on Sinhala Language. Results indicate that adequate parameter tuning to overcome structural differences existing in the language pair yields a satisfactory performance, providing valuable insight towards further research in this area.

1 Introduction

Machine Translation can be stated as an attempt to convert text from one source language to another target language using an automated procedure. The need of machine translation is visually evident day by day, with the need of overcoming the language barrier and communicating with different communities.

Machine Translation approaches range from rule based approaches to data driven approaches. Statistical Machine Translation (SMT) is a data driven approach, and is based upon statistical models which are build upon a bilingual corpora. Parameters of this models are derived based on the statistical properties of the corpora. SMT focuses on building the needed models and discovering and experimenting with different model parameters which improve the output obtained.

However, most experiments have been conducted among European languages. Efficient model generation and optimized parameter values have been calculated for those languages, leaving the way open to improvement of quality on those languages.

2 Background and Related Work

Sri Lanka is a country where three main languages are being spoken: Sinhala, Tamil and English. The deficiency observable is that most of the people who speak Sinhala or Tamil are not well aware of English: leading to barriers for information access, as well as ethnic group misunderstandings. Hence it would be a good approach to analyze how successful an application of Statistical Machine Translation be on English to Sinhala Language Translation.

Not much research has been conducted in this area regarding Sinhala Language. Weerasinghe (2004) has made a comparison in English-Sinhala translation and Tamil-Sinhala translation using SMT, arriving to the conclusion that languages which are closer in their origins do perform better in this approach. Developing an Asian-English translation is described by Ramanadan et al. (2006). An English-Tamil translator development is conducted by Germann (2001). Other attempts include translation between European languages. (Brown et al. (1990), Jones and Eisele (1983))

Section 3 and 4 would describe the basic SMT process and parameter tuning. Section 5 would describe the translation process. Experimental Setup would be explained in Section 6, followed by Results and Discussions in section 7 and the ending conclusion in Section 8.

3 The SMT Model

Given a source sentence f and target sentence e , the basic equation for getting the best possible target sentence is (Brown et al. (1983)):

$$P(e|f) = \frac{P(e)P(e|f)}{P(f)}$$

This provides the following estimation for get-

ting the maximum possible target output sentence,

$$P(e|f) = \operatorname{argmax}(P(e)P(f|e))$$

The $p(e)$ component is the language model, which takes care of the fluency of the target sentence. $p(f|e)$ is the translation model, which proves the most possible translation for the given sentence. By decoding these two models gaining of a fluent as well as a correct output is expected.

Phrase Based Translation is also being used in place of word based translation. Here, the input text would be split to phrases, and translation would take place considering phrases.

4 The Log Linear Approach

Above decoding with two models has now been improved with added feature functions, along with which additional features would be added to the decoding stage to improve the translation process, as done by Rui and Fonollosa (2005).

$$P(e|f) = \operatorname{argmax}(\lambda_m h_m(e, f))$$

h_m would be the system models: language model, translation model. However, the log linear model provides the ability to add up any feature function other than those two to improve the translation output. The weight assigned to each feature function is defined by λ_m .

5 Translation Process

Apart from data collection, the major components are the Language model, Translation Model and the Decoder.

5.1 Language Model

The language model in this case would be built in Sinhala. The experiments involved finding out the best smoothing technique (Chen and Goodman (1999)), experimenting backing off and interpolating models together, generating domain specific as well as a general language model and using perplexity to evaluate their quality. Smoothing techniques such as Good Turing Discounting, Natural Discounting, Kneyser Ney Discounting and Witten Bell Discounting has been used.

5.2 Translation Model

IBM models were generated as translation models and HMM models instead of IBM model 2 was also experimented to check their impact on Sinhala Language.

5.3 Decoder Model

Following Koehn et al. (2003), we expect to make use of phrase based translation rather than word based translation for our experiments, and intend to add additional feature functions to improve its performance.

With an aim to have one to many mappings between source and target words, bidirectional alignment translation models were generated. Different alignment strategies were experimented to check their impact on Sinhala Language: Intersection, Union, Grow-Diag-Final were generated between English and Sinhala. To cater the word order difference inherent in two languages, distortion models were generated to analyze their performance on our research.

Many additional features derived likewise were integrated with the simple language model and translation model ,via the log linear approach. These additional features are integrated in Moses (2007) toolkit. Thus, the additional feature functions used would be:

- Phrase Translation Probability $P(f|e)$
- Inverse Phrase Translation Probability
- Lexical Probability
- Inverse Lexical Probability
- Distortion Model Probabilities
- Word Penalty $\omega^{\text{length}(e)}$
- Phrase Penalty (Constant 2.718)

The decoding equation is

$$e_{best} = \operatorname{argmax}(P(F|E)P(LM)\omega^{\text{length}(e)})$$

The phrase translation probability component $p(F|E)$ would now be a compact translation model with all features integrated and weighted appropriately, which would be combined with the Language Model probability and word penalty.

To address the fact that not all words would be included in phrase pairs, integrating word based translation with phrase based translation was also experimented.

5.4 Minimum Error Rate Training (MERT)

After all the decoders are generated, MERT provides a way to optimize the weights that has been given to the feature functions in the decoding equation. MERT does this with the help of a development set, where a source language corpus with its reference translation would be used to provide the optimum weight setting for the decoder. In order to accomplish that, once the decoders were generated, MERT procedure was conducted. This was to check what the impact of MERT procedure on a Sinhala output would be.

6 Experimental Setup

6.1 Data collection and Data Preprocessing

For the language model, Sinhala Data files were gathered from the WSWS web site (www.wsws.org) (political domain), and the other was the UCSC/LTRL Beta Corpus which had data from several domains. For the Translation Model, the main data source was the WSWS web site. This contained documents in both English and Sinhala language. Data Preprocessing was conducted using script files to clean and align the data extracted from web pages.

6.2 Tools Used

6.2.1 Language Model

SRILM toolkit with its language model specific tools was used for this purpose.

6.2.2 Translation Model

IBM models including HMM model was generated using GIZA++.

6.2.3 Decoder

Moses, a beam search decoder providing ability to include additional feature functions was used for this purpose.

6.2.4 Evaluation

BLEU Metric by Papineni et al. (2001) was chosen as the evaluation metric for our experiments.

7 Results and Discussion

7.1 Language Model

The statistics of data gathered for the language models from different domains are as in Table 1. The best smoothing method was determined by calculating perplexity of relevant test sets from each domain (Table 2). Both combining data of

a specific domain together to build a large LM as well as interpolating small LMs were experimented (Table 3), from which we concluded that Unmodified Kneyser-Ney Discounting along with interpolation is the best to be used with Sinhala data.

Domain	Sentences	Unique Words
Political	17570	44652
Feature	24655	70690
Foreign	4975	24999
Other	2317	9378

Table 1: Language Model Data Statistics.

	Political	Feature	Foreign	Foreign
KN	604.46	825.61	744.11	601
UKN	584.63	763.37	712.88	582.25
UKN+INT	572.23	723.31	689.99	579.62
WB	649.02	814.46	766.61	664.13
WB+INT	651.53	799.53	767.62	662.45

Table 2: Perplexities against In-Domain Test Sets

Corpus	Interpolation Weight			
	0.0	0.4	0.5	0.8
Mixed LMs	572.23	563.35	569.81	596.62
LM from Combined Corpus	603.93			

Table 3: Mixing and Interpolating LMS

7.2 Translation Model and Decoding

Table 4 shows the statistics for parallel data gathered for building the translation model. The whole corpus was divided into training and test data sets of 50, 100 sentences accordingly.

Different alignment strategies and reordering strategies together were experimented with test sets of 50 sentences (Table 5). The reordering strategies used are msd-bd-fe, msd-bd-f, msd-fe, msd-f in these experiments.

The best configuration (7.0756) was then experimented with varying distortion limits for translations and dropping unknown words, which eventually increased the Bleu score by around 1 point (Table 6) (to around 8.11). The above found best configuration (grow-diag-final,msd-bdf, distortion limit 20, drop unknowns) was then used with the

tuning set of 2000 sentences for MERT, to get optimum weights for different features. The test set performance was again experimented with the gained optimized weights.(Table 7).

	Sentences	Unique Words
WSWS Corpus	10000	32863

Table 4: Translational Model Data

Type	msd-bd-f	msd-bd-fe	msd-fe	msd-f
grow-diag-final	7.0756	6.9899	6.8215	6.9147
grow-final	6.8838	6.8808	6.7891	6.8244
union	6.4264	6.3856	6.6708	6.3936
grow-diag-final	6.8530	6.9815	6.8705	7.0341

Table 5: BLEU Values against alignment and re-ordering strategies

DL	0	4	8	12	20	20du	25du
BLEU	3.76	6.93	7.03	7.21	7.33	8.11	8.05

Table 6: BLEU values against different distortion values and dropping unknown words

7.3 Improving Phrase with Lexical Probabilities

In order to reduce the risk of having too many unknown words during decoding, word based translation was integrated along with phrase based translation approach. A very low weight was allocated to this new feature (the value given whenever a word based translation is used - in this case we will name it as lexical penalty.) so as to give priority to phrase probabilities. Table 8 shows the resulting Bleu score for 50 sentences, and then the same thing was experimented with best distortion limits and unknown words dropped (Table 9) .

With this, the performance reached almost 14 in BLEU score (grow-final, msd-bd-f, distortion limit 20 and unknown words dropped). This shows that lexical translation probabilities add a considerable effect by adding itself as a certain word based support for the phrase based translation.

The next experiment was varying the weight/value allocated whenever a word based translation (lexical penalty) occurred, to check whether it would impact the Bleu score. Table 10

Tuning Set size	Original BLEU	After MERT
2000	8.11	9.26

Table 7: Impact of MERT on BLEU

	grow-diag-final	grow-final	union
msd-bd-fe	11.6959	6.9899	11.2667
msd-bd-f	12.3110	11.5597	6.7891
msd-fe	11.6907	10.7907	10.7641
msd-f	11.3744	10.7183	10.5767

Table 8: BLEU for configurations with added lexical probability

clearly shows that the value given for the lexical penalty has played a considerable role in the Bleu value, resulting in a final Bleu score of 15.06. This shows that integrating word based translation along with phrase based translation with certain lexical penalties do impact the translation performance in a very positive manner, though the current analysis is not enough to successfully point out that this specific value would be good to be used as the lexical penalty value in English to Sinhala Translation.

8 Conclusion and Future Work

The purpose of this research was to find out how English to Sinhala Translation can be accomplished using SMT techniques. Alignment models, reordering models, integrating phrase models with word models and distortion limits together built up a considerably significant increase in performance. All these add up to the conclusion that the structural differences inherent in these two languages can be overcome to a certain extent via appropriate parameter tuning. This is specially visible via the increase in Bleu score upon varying reordering models and distortion limits. Previous research (Weerasinghe , 2004) turned out to provide a Bleu score of around 2-6, where as this research yielded a Bleu score of around 15 , symbolizing an impressive insight on future work in this area.

Possible future work might involve using larger corpora for Sinhala Language. Another possible research would be to find out better phrase extraction heuristics for the phrase based translation model, so that the decoding process relies on the word based model as less as possible. This would make the translation process be relying more on phrase based translation, which would eventually

	Grow		grow-final		union	
	dl20	+du	dl20	+du	dl20	+du
msd-bd-fe	12.73	13.55	13.36	13.66	12.39	12.31
msd-bd-f	13.27	13.69	13.75	13.99	12.23	12.32
msd-fe	12.63	13.14	12.99	13.23	11.08	11.29
msd-f	12.43	12.91	13.35	13.18	11.10	11.74

Table 9: BLEU Score for phrase tables with added lexical probabilities : DL20

Corpus	Lexical Penalty Value			
	0.001	0.005	0.01	0.2
Test Set 50 Configuration	13.99	14.49	15.06	13.84
Test Set 100 Configuration	11.16	11.76	11.62	11.43

Table 10: Bleu Scores for best decoder configuration with varying lexical penalty : DL20

result in better and fluent translation outputs.

To conclude with, this research can be stated as a promising start towards better information access for rural communities via English to Sinhala Machine Translation. This resesarch can be conducted to improve the performance even more by finding out other possible features which can be added to this process. In a more general point of view, this research can also be stated as an extensive research on machine translation between structurally dissimilar languages.

Acknowledgment

Authors would like to acknowledge the Language Technology Research Laboratory of University of Colombo School of Computing, Sri Lanka for providing human and physical resources to carry out this research.

References

- Brown, P. F., Cocke, J., Della-Pietra, S. A., Della-Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer R. L. and Roossin, P.S. 1990. A Statistical Approach To Machine Translation. *Computational Linguists*.
- Brown, P.F., Della-Pietra, S. A., Della-Pietra, V. J. and Mercer, R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation *Computational Linguistics*.
- Chen, S. and Goodman, J. 1999. An empirical study of smoothing techniques for language modelling. *Computer Speech and Language*,

Germann, U. 2001. Building a Statistical Machine Translation System from Scratch: How Much Bang Can We Expect for the Buck? *Proceedings of the Data-Driven MT Workshop of ACL-01, Toulouse, France*.

Jones,D. and Eisele, A. 2006. Phrase-based Statistical Machine Translation between English and Welsh *International Conference on Language Resources and Evaluation, 5th SALT MIL Workshop on Minority Languages: Strategies for developing machine translation for minority languages, Genoa, Italy*.

Koehn, P. , Och, F. J.and Marcu, D. 2003. Statistical Phrase Based Translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada*.

Moses. 2007. Moses - a factored phrase based beam-search decoder for machine translation. <http://www.statmt.org/moses/>,(cited 2011.01.10).

Papineni, K. A., Roukos, S., Ward, T. and Zhu, W.J. 2001. Bleu : A method of automatic evaluation for machine translation. *Publications Manual*. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Ramanadan,A. , Bhattacharyya, P., Sasikumar, M. and Shah, R. 2006. Some Issues in Automatic Evaluation of English-Hindi MT: More Blues for BLEU.

Rui, M. and Fonollosa, A. 2005. Improving Phrase-Based Statistical Translation by modifying phrase extraction and including several features *Proceedings of the ACL Workshop on Building and Using Parallel Texts,Ann Arbor:June 2005.Association for Computational Linguistics*.

Weerasinghe, A. R. 2004. A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation. *SCALLA*.

Strategies for Research Capacity Building in Local Language Computing: PAN Localization Project Case Study

Sana Shams

Center for Language Engineering,
KICS, UET, Lahore. Pakistan
sana.shams@kics.edu.pk

Dr. Sarmad Hussain

Center for Language Engineering,
KICS, UET, Lahore. Pakistan
sarmad@cantab.net

Abstract

Strengthening indigenous research capacity versus technology transfer (Harris, 2004; Nokolov and Illieva, 2008) is the most effective process for advancing research, specifically in localization. This paper discusses different strategies undertaken by the PAN Localization project in building research capacity for localization in the partner countries based on principles of capacity building defined in literature.

1 Introduction

The use of ICTs is often accentuated as the most promising and fundamental driver of social and economic development (Lutz, 2003). This is because of its potential to enable the underprivileged to leapfrog the barriers to information access, improve livelihood opportunities and communicate with people across the globe (World Bank, 2002). However, only 21% of developing economies can access information on the internet (ITU, 2011). This digital divide is largely perpetuated due to English, being the lingua franca for ICTs (Pimienta, 2005). This language based digital divide is very pronounced in Asia, where about 2,322 languages are spoken (Lewis, 2009) and only 2% know how to read and write in English, a pre-requisite for ICT usage.

Local language computing also called localization or enabling ICTs in a local language is essential for effective ICT use in Asia (Gul, 2004). This paper presents strategies and interventions by PAN Localization project¹ for building sustainable research capacity in localization, through partnership of 11 Asian countries, since 2004. The paper is organized as follows: Section 2 defines localization and its

related knowledge areas, Section 3 presents capacity building framework and derives the structural levels and principles upon which capacity enhancement initiatives must focus, and Section 4 presents the PAN Localization project and the capacity building interventions that have been carried out through the project following each of the defined capacity building principles. Based on the findings, Section 5 recommends the capacity building strategy. Section 6 concludes the paper.

2 Localization Research

As defined by Hussain and Mohan (2007), localization is “*The process of developing, tailoring and/or enhancing the capability of hardware and software to input process and output information in the language, norms and metaphors used by the community.*” It is a three step process. First, the linguistic analysis is required to document (and standardize) language conventions that are to be modeled. Second, localized applications (both basic and intermediate level) e.g. fonts, keyboard, locale, spell checkers, etc. need to be developed to enable basic input and output of text in a local language. Thirdly to provide comprehensive access and assist content development, advanced applications like translation systems, speech dialogue applications, etc., need to be developed.

Localization therefore requires significant knowledge of linguistics (phonetics, phonology, morphology, syntax, semantics and pragmatics), signal and speech processing, image processing, statistics, computational linguistics and advanced computing (Hussain et al, 2007). This research being language dependent, entails nurturing indigenous *research capacity* (Breen et al, 2004, DFID, 2007) at the levels of individuals, organizations, and systems to sustain.

¹See www.PANL10n.net.

3 Research Capacity Building Models

Research capacity building (RCB) frameworks define levels and set of practices that help build capacity. RCB frameworks available in literature (Cooke 2005, Neilson and Lusthaus 2007, Wignaraja 2009) largely recommend three structural levels and six basic principles upon which capacity building must be designed as discussed in the sub-sections below.

3.1 Structural Levels of RCB

Structural level of RCB defines the point of view upon which capacity development initiatives must be targeted. They include individual, organizational and system levels (Neilson and Lusthaus 2007, Breen et al, 2004). Some frameworks follow a hierarchical categorization of these levels (Potter and Brough, 2004) and others form a phase-wise development plan (Wibberley, Dack, & Smith, 2002, Breen et al, 2004) where capacity building at certain prior level necessitates capacity development at the next level. Interventions however cannot be carried out at a certain level in isolation. Every activity accomplished at a certain level has impact on the other levels.

3.2 Principles of RCB

Cooke (2005) recommends six principles of capacity building that include focusing interventions on: skill development; focus on close to practice research; establishment of linkages, partnerships and collaborations; development of capacity for dissemination and impact; sustainability and continuity of research and development of infrastructure. Each principle is briefly described below.

3.2.1 Skill Development

RCB requires a multi-faceted skill development process through training and supervision to primarily develop technical, managerial, and publishing skills (Harris 2004, Raina 2007). Skill development can also be viewed in the context of career development and generating opportunities to apply research skills in practice (Rhee and Riggins. 2007).

3.2.2 Training on Close to Practice Research

A foremost principle of RCB is in directing researchers' ability to produce research that is useful for informing policy and practice (Cooke,

2005). Thus capacity building interventions ensure that research is "close to practice" such that new knowledge generated can directly impact development.

3.2.3 Development of Linkages

Developing linkages, partnerships and collaborations is a reciprocating process of involving organizations in the knowledge information chain for fostering development and diffusion of quality research (Wignaraja 2009, Breen et al 2004). It also harnesses an increased knowledge base for research development and enhancement.

3.2.4 Dissemination and Impact

Dissemination of research, through peer reviewed publications and presentations at academic conferences, is essential for sharing knowledge (Harris 2004, Breen et al 2004). Capacity building for wider research dissemination incorporates instruments of publicity through factsheets, the media and the Internet (Cooke 2005) for a variety of stakeholders, including public, policy makers and the relevant research community.

3.2.5 Sustainability and Continuity

RCB must ensure strategies for maintenance and continuity of the acquired skills and structures to undertake research. Wignaraja (2009) defines capacity development as a process of transformation that emerges from within the individuals, organizations and systems. Long term sustainable capacity development requires consolidation of local systems and processes through practice.

3.2.6 Infrastructure Development

Rhee and Riggins (2007) defines infrastructure as a set of structures and processes that are set up to enable the smooth and effective running of research projects. These include availability of technical resources including equipment, books, connectivity, etc. as well as sound academic and managerial leadership and support for developing and sustaining research capacity.

Based on the above categorization, the following section describes strategies undertaken by PAN localization project 2004-2013 for fostering localization research capacity in the partner countries.

4 PAN Localization Project

PAN Localization project (www.pan110n.net) is a regional initiative to develop local language computing capacity in Asia. The project involved partnership of 21 organizations across Afghanistan, Bangladesh, Bhutan, Cambodia, China (Tibet Autonomous Region), Indonesia, Laos, Nepal, Mongolia, Pakistan, Sri Lanka, for conducting research on fifteen local languages spoken across developing Asia.

4.1 Strategies for Localization Capacity Building

Prior to initiating the capacity development program, baseline study of the existing research capacity in partner countries was conducted to help in devising the strategies (Hussain, 2004). The study showed that while teams had very limited experience in basic localization (except in a couple of countries) and many countries had no work done in localization policy development and development of intermediate and advanced localization. Also there was hardly any experience in inter-disciplinary research and development across computing, engineering and linguistics. Regarding team management, only two country team leaders had experience in running long-term multi-disciplinary projects.

Faced with the above capacity challenges, appropriate measures had to be undertaken, focusing across the six principles of research capacity building to target holistic improvement. Specific interventions for each of the principle are discussed in the section below.

4.1.1 Skill Development

Skills development through the project has been focused on building technical skills to conduct and publish localized research outputs. Strategies practiced in this context are explained below.

4.1.1.1 Undertaking Localization Research

Foremost strategy employed for technical skill development has been to require each project team to deliver specific research outputs. This strategy served as a persistent capacity building process that enabled researchers to work on real problems and find research solutions through involvement in problem identification, project designing, implementation, quantitative and qualitative analysis. Working directly to address the technical, linguistic, social and managerial challenges also developed lasting confidence to

undertake further work in the future. The following table presents the comparative figures for assessment of their teams' capacity by the project leaders from each partner country. The levels are on a scale of 1-5, collected at the beginning of project Phase 2 in 2007 and towards the end of this phase in 2009. These figures are derived from a more detailed survey done for the 11 partner countries.

Capacity Building Target Area	Year 2007	Year 2009
Project development	2	5
Project design	3	5
Problem identification	3	5
Project implementation	3	5
Analysis Ability	3	4
Communication Ability	2	4
Multi disciplinary research Ability	2	4
Quantitative analytical skills	2	4
Qualitative analytical skills	2	4

Table 1: Progression of Capacity Building of Teams in Local Language Computing

This process also trained the researchers to work in multi-disciplinary team comprising of computer scientists, linguists, sociologists, stenographers, within each partner. Where the project teams were challenged and stopped short of skills to meet the planned research outputs, short term training or mentor placement programs were initiated as explained in the following section.

4.1.1.2 Short Term Training

Short term training was a skill building strategy that was designed normally as a week-long activity targeting training on a certain research area. In addition to building individual's capacity, this strategy would also help build institutional capacity. Trainees receiving the short term training were not limited to project staff only but would also include additional relevant staff where this training was organized. Six short term training were conducted during the project, covering a varied set of topics, for example, FOSS localization, OCR development,

linguistics and monitoring and evaluation using outcome mapping framework.

4.1.1.3 Mentor Placement Program

Where the country team required longer training to address capacity challenges, mentor placement programs were initiated, which provided technical and management support to partner countries. Two different models have been adopted in this context. In first model (referred to as mentor Placement I in Table 2), a mentor from within the partner countries was sent to partner needing support. Three such mentor placements were conducted from 2004-2007, and 2 were held during the second phase of the project. In second model (referred to as mentor Placement II in Table 2) respective country component nominated one or two persons from team to stay with mentoring organization for the training duration. One such placement was initiated in the project's first phase of the project, while 5 such placements were done in the Phase 2. Both models have been worked out equally well. An extension of first model has also been tried by providing the remote mentoring facility after completion of training, which has also proved effective in achieving the research outcomes.

4.1.1.4 Summer School in Local Language Computing

The project further initiated an innovative form of technical training called Summer School in local language computing. This was a semester equivalent (three month long) extensive academic program with credit for five graduate courses in linguistics and computational linguistics that were not offered in the partner countries. The credit hours earned through the semester were transferrable in any other graduate program. The course instructors selected to teach these courses were experts in their fields chosen from around the world. This helped quickly boost the capacity of the partner teams, enabling the transition from undertaking research in localization in Phase 1 to more advanced research in language computing in Phase 2.

4.1.1.5 Support for Higher Studies

As a strategy for continued and advanced training in local language computing, the project provided completed or partial scholarships for many team members for pursuing higher studies in disciplines related to localization research. Specific researchers were funded in Bangladesh,

Cambodia, Indonesia, Mongolia, Pakistan and Sri Lanka, to accomplish their academic research through working on Project. In addition, project facilitated these team members by providing time for studies and examinations and in certain instances by supporting the tuition fee for their degree program. This support for higher studies was also used as a strategy for retention of researchers, as these team members would remain with the Project until degree completion.

4.1.1.6 Presentation at Workshops and Conferences

A number of researchers from different partner countries participated and presented their work at national and international workshops and conferences. This was a testimony of the maturity of their research skills developed during the project. As an incentive for producing publishable research the project supported their travel and participation expenses.

The following table summarizes the number of times each type of strategy is employed during the respective project phases.

Training Strategy	Phase 1 (2004-07)	Phase 2 (2007-10)
Short Term Training	6	-
Mentor Placement (I)	3	2
Mentor Placement (II)	1	5
Summer School	1	-
Support for Higher Studies	5	9
Conference Participation	12	40

Table 2. Capacity Building Interventions During Project Phases 1 and 2

The table shows that Phase 1 focused on short training and mentor placements I. As the teams had acquired reasonable competence, during Phase 2 strategies targeted collaborations, e.g., Mentor Placement II, summer school and conference participation, and longer term impact, e.g. through support for higher studies.

4.1.2 Training to Conduct Close to Practice Research

Research work is more useful if its outputs can directly provide socio economic benefit to the communities. Following this motivation, the project instituted the following strategies.

4.1.2.1 Partnerships for Outreach

An outreach component for the project research work was specifically implemented with most of its project partners during the second phase of the project, while the first phase had focused on development of the technology. For this purpose, in the second phase the project also developed partnerships with civil society organizations to specifically focus on dissemination of technology to end users in the partner countries, with explicit funding allocations to support the partnerships. For example, Nepalinux developed by Madan Puraskar Pustakalaya (MPP) was used by E-Network Research and Development (ENRD) to train five rural communities around Nepal, which included farmers, mothers' group, and retired army men. They used the Nepali language applications to communicate with their relatives abroad and to develop online community portals. Similarly, Pakistan component collaborated with District Governments of Sargodha, Chakwal and Attock to deploy localized open source applications in ten rural schools, training more than 200 school students and teachers on information access, communication and content generation.

The partnerships have enabled partners focusing on outreach to appreciate technical challenges and helped the technical partners to appreciate the end user dissemination and adoption challenges. Both lessons significant for planning research they would undertake in the future.

4.1.3 Development of Linkages

Research groups often operate in isolation, limiting the scope and success of their work. The project has been focusing on collaborative learning. Experiences of researchers who are working successfully under similarly resource-constrained conditions engender trust and motivation. To ensure the project teams are well knit into the localization domain, the project implemented multiple strategies.

4.1.3.1 Inter-Disciplinary collaborations within teams

As local language computing requires research in linguistics, technology and user adoption, partners were encouraged to develop multi-disciplinary teams both at advisory and implementation levels. This included computer science professional working directly with linguists and sociologists, these researchers coming from a very different work cultures. Many of the partner countries did not have such collaborations earlier and thus developed individual and institutional capacities and also larger context, where the conventional circles for the communities could see possibilities and benefits accrued.

4.1.3.2 Inter-Disciplinary Collaborations Across Teams

Partner teams were encouraged to establish partnerships and collaboration with institutions that had more expertise in a specific field. These collaborations enabled the partners to collectively plan the technical and financial details, exchange data and technology and discuss and formalize shared intellectual property regimes, building institutional capacities in the context. For example, in Bhutan, Department of IT, the primary partner institute of the PAN L10n project collaborated with Dzongkha Development Authority, their national language development and language standardization authority to develop the technical terminology translations for the software. The advantage of those collaboration was that once the terminology was developed by DDA, it would become a national standard for such terminology translation. In Cambodia the PAN Cambodia collaborated with Institute of Technology, Cambodia (ITC) that had professors working on localization research and students taking up localization research projects in their BS final year projects. Such examples were practiced in all partner countries.

4.1.3.3 Regional and International Collaborations

As a strategy to develop international collaborations, the project has been organizing regional training, conferences and workshops, in which experts from the region are invited. These have provided opportunities to meet and discuss opportunities for collaboration. As a salient example, project partners have been interacting

with NECTEC, Thailand, which have eventually resulted in formal bi-lateral and multi-lateral partnerships. The project has also worked with researchers from Korea, India, Japan and regional organization like Asian Federation of Natural Language Processing. Such interactions have also resulted in direct partnerships between Microsoft and country partners resulting in the development of Language Interface Packs (LIP) for MS Office and Windows in Urdu, Pashto, Bangla, Sinhala, Khmer, and Lao, by the project partner countries.

4.1.3.4 Online Research Networks

The project teams have been participating in online research networks, discussion groups, communities and forums for collaboration, knowledge sharing and learning. The work they have performed have given them confidence not only to learn but also contribute on these online forums. The project created an online support network to encourage project partners to be a part of an online learning culture. The project partners have been participating on this forum, sharing their project experiences with each other. At the beginning of the project, this network was enrolled by 11 researchers with grew to 110 researchers from 25 different countries by the end of project phase 1 in 2007. Nepal and Bangladesh team discussed their challenges in developing spell checker for open source software for Brahmic scripts. The solution based on HunSpell by Nepalese helped the team develop Bangal spell checker in Bangladesh.

4.1.4 Dissemination and Impact

Dissemination is an essential part of undertaking research. The project used a variety of strategies to focus on distributing its outputs to a wide variety of stakeholders. Some of the activities are described below.

4.1.4.1 Project Websites

The main and sustained source of information and outputs of the project has been the project website. The core site has been maintained by the project's regional secretariat, though the project required each country team to nominate one person from their team to act as a website coordinator and provide local content for the centrally maintained multilingual website www.pan110n.net. In addition the teams also hosted their separate websites providing detailed information about their respective research

groups, hosted by their organizations, that are linked from the main website as well. This has given global access to project outputs.

4.1.4.2 Awareness Seminars

The project has organized awareness seminars to disseminate and publicize research results to local community. These seminars have been attended by a large number of participants from academia, public and private sectors. Through these seminars partner institutions have been regularly presenting their work to the key stakeholders from government, IT industry, academia, media, and end user communities. 4 such seminars were conducted in project phase 1, while there are funds for conducting 16 more seminars by the end of project's phase 2.

4.1.4.3 Promotional Material

Development of promotional material has been an integral strategy for research dissemination. In addition to publicity project flyers, teams have distributed CDs containing project outputs such as NepaLinux, Dzongkha Linux. For example, an estimated 3000 copies of CDs/DVDs have been distributed to various stakeholders. The videos about the project have also been produced and uploaded online for global audience.

4.1.4.4 Participation in Events and Competitions

Many of the project outputs have been presented at national and international forums and have also been awarded. For example, NepaLinux has been displayed at annual exhibitions like Info Tech organized by the Computer Association of Nepal (CAN) for the last four years. The national and international awards, including APC Chris Nicol FOSS Prize, Manthan awards, have also contributed to propagating the work to relevant research communities.

Project partners have been involved in designing, developing and disseminating the material developed, which has contributed to mutual capacity to disseminate research.

4.1.4.5 Planning and Evaluating Impact

Six of the eleven country partners initiated an outreach component of research in their projects, gaining insight on challenges in technology adoption and experience in designing, implementing and evaluating such interventions, also developing their capacity to reflect on

technology design and implications in the context. The focus was firming up by explicit training of the partners to plan, monitor and evaluate impact on society using formal framework of Outcome Mapping (Earl *et al.*, 2001). There was also an explicit focus on Gender, developed with collaboration with Gender Evaluation Methodology (GEMII) project, which resulted in the development of the Gendered Outcome Mapping Framework (Shams *et al.* 2010), practiced in the program. An online tool has been developed to put the training into practice and is being used to collate and publish data regarding the project's outcome challenges and progress markers.

4.1.5 Sustainability and Continuity

PAN Localization project has taken specific measures at individual, organizational and policy levels for building sustained indigenous research capacity to carry out local language computing in partner countries. Some of the specific strategies undertaken in this regard have been as follows.

4.1.5.1 Building Indigenous Localization Research Capacity

The project has focused on development of indigenous human resource capacity for localization by engaging twenty one partner organizations, in eleven different countries, working on fifteen different local languages. A significant number of technical developers, linguists and social scientists have been trained through the project. The following table gives the numbers of people engaged and trained through the project.

	Mgmt	Tech	Lang	Social Sc.	
Af	2	2	3	0	7
Bd	3	10	3	6	22
Bt	1	8	4	0	13
Kh	7	39	13	3	62
Cn	4	11	3	0	18
Id	3	3	12	0	18
La	5	9	6	3	23
Mn	3	15	9	0	27
Np	6	12	4	9	31
Pk	6	27	7	4	44
Sl	3	7	3	1	14
	43	143	67	26	279

Table 3. Number of Researchers in the Project

4.1.5.2 Advancing Research Capacity

For sustainability and advancement of localization research, country projects were housed within universities and government institutions that would continue the research beyond the project duration. Dedicated research centers were established through the project, for example. Center for Research in Bangla Language Processing, in Bangladesh, Language Technology Research Center in Sri Lanka, R&D Division within Department of IT, Bhutan, Language Research group in National Agency for Science and Technology, Laos, Language Technology Research Lab at National University of Mongolia, Center for Language Engineering at University of Engineering and Technology, Pakistan and Language Technology Kendra in Nepal. This has instigated further localization collaboration and research. The following table presents the localization research outputs successfully implemented in the partner institutes showing the research maturity that they have gained through the project.

	Basic		Inter-mediate		Advanced	
	Ph 1	Ph 2	Ph 1	Ph 2	Ph 1	Ph 2
Afghanistan	*	*		*		
Bangladesh	*		*	*	*	*
Bhutan	*	*	*	*		*
China		*		*		*
Cambodia	*	*	*	*		*
Indonesia						
Laos	*		*	*		*
Mongolia		*		*		*
Nepal			*			*
Pakistan		*		*		*
Sri Lanka	*		*	*	*	*

Table 4. Research Outputs of Country Teams in Project Phase I & II

4.1.5.3 PAN L10n Multilingual Chair in Local Language Computing

To sustain and consolidate the regional momentum of localization research capacity building initiated through the project, a permanent research chair for multilingual computing has been established at the project

regional secretariat in Pakistan funded by International Development Research Center (IDRC), Canada. Establishment of this research chair would provide a sound foundation to sustain, nurture and grow the network of localization researchers, and to provide direct support in language computing community, including researchers and policy makers.

4.1.6 Infrastructure Development

Lack of appropriate infrastructure for conducting research including equipment, books, journals and inability to support the recurring administrative expenses may become an impediment to conduct scientific research. Thus appropriate localization research infrastructure has also been established at partner countries. In addition the project also provided support for buying books and journals, and specialized software for localization research in the partner countries. The following table summarizes the interventions conducted by the project at each of the structural levels.

Principles of RCB	Ind.	Org.	System
Skill Development	4.1.1.1	4.1.1.2	*4.1.1.6
	4.1.1.2	4.1.1.3	*4.1.3.3
	4.1.1.3		
	4.1.1.4		
Close to Practice Research	4.1.2.2	4.1.2.1	
		4.1.2.2	
Development of Linkages	4.1.3.1	4.1.3.1	4.1.3.2
	4.1.3.4	4.1.3.2	4.1.3.3
Dissemination and Impact	4.1.4.2	4.1.4.2	4.1.4.1
			4.1.4.2
			4.1.4.3
			4.1.4.4
Sustainability and Continuity	*4.1.1.5	4.1.5.1	4.1.5.3
		4.1.5.2	
Infrastructure Development		4.1.6	

Table 5. RCB Interventions at Structural Levels

Interventions marked with asterisk (*) in the table above signify that project strategies had a cross-cutting effect across all structural levels. E.g. strategy for supporting higher studies (4.1.1.5) not only served to build their skills but also helped to sustain the researcher at individual level

5 Discussion

Though the six principles of RCB holistically address the challenge of RCB in localization, however appropriate sequencing within the six principles must be done in order to foster maximal impact (Potter & Brough, 2004). Based on the project experience RCB for localization must follow a developmental cycle within the defined capacity building principles.

Initially *Skill Building* and *Infrastructure development* must form the focus of RCB interventions for localization. Different countries require different research skills and infrastructure needs owing to the existing competencies. Thus a participatory need analysis should be performed to ensure skill development is based on national priorities and capacity. Based on the identified RCB needs, appropriate mentorship structure and organizational resources may be planned to ensure development of the research base.

Secondly, localization RCB can be built to carry out *close to practice research*, of direct benefit in practice, after development of technology the basic and intermediate levels of localization research as a pre-requisite, only then can the research be conducted that harness solutions that can be readily used by the benefitting populations. At the same time, capacity building initiatives must target to form *linkages and partnership* with relevant academic, policy making, regional standardization bodies, and public and private sector bodies. This would follow skill development at level 1 as synergetic and mutually benefitting collaborations can only be developed if the local teams are able to contribute back to the knowledge network.

Finally, *research dissemination and sustainability* must be targeted as these RCB dimensions provide research maturity to publish in technical forums, and compete for funding.

6 Conclusion

ICT human resource capacity indicators signify a steep demand for localization skills in ICT professional with the increasing ICT diffusion in the Asia Pacific (Rhee and Riggins, 2007, Raina, 2007). UN-APCICT/ESCAP (2010) speculates that existing institutions for ICT education and training in the regional cannot fulfill this demand. Therefore localization RCB must be taken up as a national and regional priority in

order to bridge the demand supply gap of the required localization RCB in developing Asia.

References

- Bali moune-Lutz, Mina. 2003. *An Analysis of the Determinants and Diffusion of ICTs in Developing Countries*. Information Technology for Development 2003, 10:151–169
- Breen, C. M., Jaganyi, J. J., Van Wilgen, B. W., and Van Wyk, E. 2004. *Research Projects and Capacity Building*. Water SA, 30(4):429-434.
- Cooke, Jo. 2005. *A Framework to Evaluate Research Capacity Building in Health Care*. BMC Family Practice, 6(44)
- Department for International Development (DFID). 2008. *DFID Research Strategy 2008-2013. Working Paper Series: Capacity Building*. Accessed from http://www.dfid.gov.uk/r4d/PDF/Outputs/Consultation/ResearchStrategyWorkingPaperfinal_capacity_P1.pdf
- Earl, Sara., Carden, Fred. & Smutylo, Terry. 2001. *Outcome Mapping: Building Learning and Reflection into Development Programs*. Ottawa, Canada: International Development Research Centre
- Gul, Sana. 2004. *Dilemmas of Localisation in Asia*. I4D Online, 2(12)
- Harris, Eva. 2004. *Building Scientific Research Capacity in Developing Countries*. European Molecular Biology Organization Reports 2004, 5(1):7-11
- Hussain, Sarmad. 2004. *Developing Local Language Computing*. I4D Online, 2(6)
- Hussain, Sarmad and Gul, Sana. 2004. *Localization in Pakistan*. Localisation Focus: An International Journal for Localization, 3(4)
- Hussain, Sarmad, Gul, Sana and Waseem, Afifah. 2007. *Developing Lexicographic Sorting: An Example for Urdu*. ACM Transactions on Asian Language Information Processing (TALIP), 6(3)
- Hussain, Sarmad and Mohan, Ram. 2007. *Localization in Asia Pacific*, in Librero, Felix (eds), Digital Review of Asia Pacific, 2007-08 Sage Publications India Pvt. Ltd.
- International Telecommunication Union (ITU), 2011. *The World in 2010, ICT Facts and Figures*. Accessed from <http://www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf>
- Lewis, M. Paul. 2009. *Ethnologue: Languages of the World*. SIL International.
- Neilson, Stephanie and Lusthaus, Charles. 2007. *IDRC Supported Capacity Building: Developing a Framework for Capturing Capacity Changes*. Universalia.
- Nikolov, Roumen and Illieva, Sylvia. 2008. *A Model for Strengthening the Software Engineering Research Capacity*, SEESE'08, Germany.
- Pimienta, Daniel. 2005. *Linguistic Diversity in Cyberspace – Models for Development and Measurement*, in UNESCO Institute for Statistics (eds), Measuring the Linguistic Diversity on the Internet, UNESCO
- Potter, C. and Brough, R. 2004. *Systemic Capacity Building: A hierarchy of Needs*. Health Policy & Planning, 19(5):336-345.
- Raina, Ravi. 2007. *ICT Human Resource Development in Asia and the Pacific: Current Status, Emerging Trends, Policies and Strategies*. UN-APCICT. Accessed from <http://www.unapcict.org/ecohub/resources/ict-human-resource-development-in-asia-and-the>
- Rhee, Hyeun-Suk and Riggins, Frederick J. 2007. *Development of a Multi-Factor Set of Country-Level ICT Human Resource Capacity Indicators*. UN-APCICT. Accessed from <http://www.unapcict.org/ecohub/resources/development-of-a-multi-factor-set-of-country-level>
- Shams, Sana, Hussain, Sarmad. and Mirza, Atif. 2010. *Gender and Outcome Mapping*, in Belawati, T. and Baggaley, J. (eds) Policy and Practice in Asian Distance Education, Sage Publication India Pvt. Ltd.
- United Nations Asian and Pacific Training Centre for Information and Communication Technology for Development (UN-APCICT/ESCAP). 2010. *ICT Human Capacity Building for Development*. UN-APCICT-ESCAP 2009.
- Wibberley, C., Dack, L. M. F., and Smith, M. 2002. *Research-minded Practice in Substance (mis) Use Services*. Journal of Substance Use, 7(1):19-23.
- Wignaraja, Kanni. 2009. *Capacity Development: A UNDP Primer*. Accessed from http://content.undp.org/go/cms-service/download/asset/?asset_id=2222277
- World Bank. 2002. *Information and Communication Technologies: A World Bank Group Strategy*. World Bank Group.

Information Extraction and Opinion Organization for an e-Legislation Framework for the Philippine Senate

Allan Borra

Charibeth Cheng

Rachel E. O. Roxas

Sherwin Ona

Center for Language Technologies &

Center for ICT for Development

College of Computer Studies

De La Salle University, Manila, Philippines

{borgz.borra, chari.cheng, rachel.roxas, sherwin.ona}@delasalle.ph

Abstract

This paper outlines the Language Technologies (LT) used for an e-Legislation Framework prototyped for the Philippine Senate's Committee on Accountability of Public Officials and Investigations (or Blue Ribbon Committee). The e-Legislation system uses an e-Participation framework of having both top-down (or government providing information to citizenry) and ground-up empowerment (or citizens participation). The Language Technologies employed manage the information obfuscated in unstructured text coming from both directions mainly for the purpose of policy-making. The top-down component utilizes a conventional Document Management System augmented with Information Extraction that allows for better and almost instantaneous management of information from uploaded documents. The ground-up component uses an online forum scheme and augmented with Automatic Opinion Classification and Clustering. Both e-Participation framework components (top-down and ground-up) are integrated in a single portal. This paper focuses on the technical issues of the language technologies used: information extraction and opinion classification with data clustering. Preliminary testing and results are discussed to which the information extraction performed 95.42% accuracy while the opinion organization consisting of the detection, classification and clustering modules have accuracy rates of 50%, 50.5% and 53.85%, respectively.

1 Introduction

The increase use of ICT in different sectors makes it a viable medium for e-Government and e-Participation. Macintosh (2007) outlined an e-Participation framework as shown in Figure 1. As shown, e-Participation has two aspects: top-down and ground-up. The interplay of the two compo-

nents is vital in sustaining the whole e-Participation framework. Transparency and pushing of information of government empowers citizenry to participate. Empowered citizenry's active participation may lead to good government and governance, as well as towards crafting of more pertinent policies. The main medium between these two components is texts and language that resemble in conversations and documents. The main goal is to structure the information from unstructured text coming from both directions.

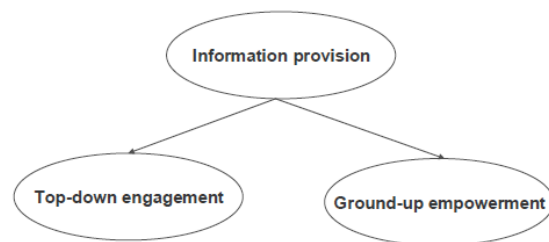


Figure 1. e-Participation Framework by Macintosh (2007)

An e-Legislation portal was, thus, developed to have both components of top-down engagement and ground-up empowerment involved. Describing Figure 2, an open-source Document Management System leverages the top-down (government to grassroots) pushing of information to citizenry, while an online forum scheme was implemented to leverage the ground-up empowerment by allowing for citizenry (or netizens) to actively participate, post opinions and comments, and interact with government and other citizenry. As documents and information being pushed, as well as netizens' opinions and comments, increase in size and magnitude, information get obfuscated more easily, especially since the main sources of information are in texts within documents, comments and opinions.

These reiterate the need to structure the information found in these unstructured texts coming from both components. This is where the portal utilize Language Technology tools to augment the two components and structure the information and open possibility to facilitate policy-making and interaction, information retrieval, and may even open up creation of new information from the structured data.

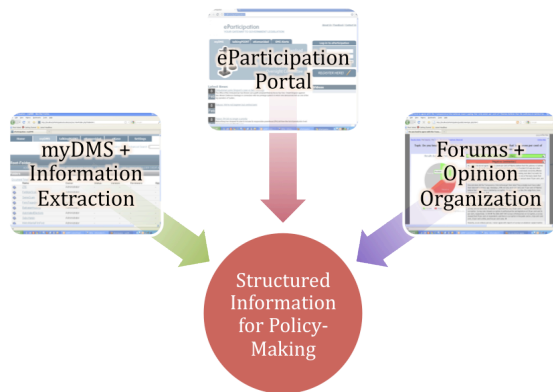


Figure 2. ICT Tools for e-Participation Utilizing Language Technology

2 Information Extraction + Document Management System = Knowledge Management

Currently, there is no electronic document management system infrastructure in the Blue Ribbon Committee (BRC), specifically in the Blue Ribbon Oversight Office Management (BROOM) of the Philippine Senate. In fact, only one person is tasked and knowledgeable of the agency's documents filing, cataloguing and retrieval procedures. The study, therefore, had a layer of modeling the business rules and process to implement the document management system (shown in Figure 3) before information extraction research were conducted and implemented. Although the whole experience of the business process modeling is a very related discourse, the focus of this section is on the information extraction and the technical aspect of the technology.

Information extraction is the process of transforming unstructured information of documents into a structured database of structured information. The underlying architecture is based on Hobb's (1993) Architecture: text zoning, pre-processing, filtering, pre-parsing, parsing, fragment combination, semantic interpretation, lexical disambiguation, co-reference resolution, and template generation. Modifications to the architecture, such as the sequence and functions of

modules, were done to address the idiosyncrasies of the documents available in the government agency.

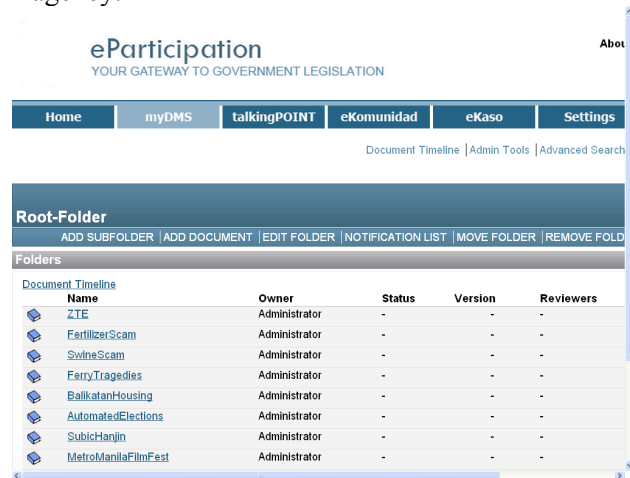


Figure 3. Document Management System Screenshot

2.1 System Architecture

The e-Legislation information extraction architecture can process different types of Blue Ribbon Committee documents. Although the Document Management System can handle any file types, only documents that manifest a regular format are processed. These documents are hearing highlights, hearing invitations, senate memorandums, documented evidences, requested administrative documents, complaints, endorsements, referrals, notification or notice of hearings and committee reports. As a result, the system handles different templates for each type of document. Considering this, the system's semantics would still be able to differentiate what template to use for a specific document.

Figure 4 shows the architecture of the system. The modules are the preprocessor, pre-parser, semantic tagger, co-reference resolution, template filler and evaluation. Under the preprocessor, there are 6 submodules: tokenizer, sentence splitter, cross-reference, part of speech tagger, unknown word and named entity recognition.

In a nutshell, a document undergoes initially the Pre-processing stages which undergoes:

1. Sentence Splitting, which removes headers and breaks down input document into series of sentences;
2. Tokenizing, which simply breaks down sentences and detects word boundaries;
3. Cross-referencing, which further sifts through the tokens and looks for entities (names) in the sentences following Schwartz and Hearst (2003) acronym detection;

4. Part-Of-Speech (POS) Tagging, which annotates the tokens within the sentence with appropriate Part of Speech Tags using LingPipe (2003);
5. Unknown-Word Detection, which classifies words that are unclassified or unknown from the POS Tagger process. It uses the ANNIE POS Tagger (1995) to represent the unknown words and classify them for post processing; and
6. Named Entity Recognition, which uses LingPipe's (2003) Dictionary Mapping named entity recognition or a look-up table dynamically added by the cross-reference phase;

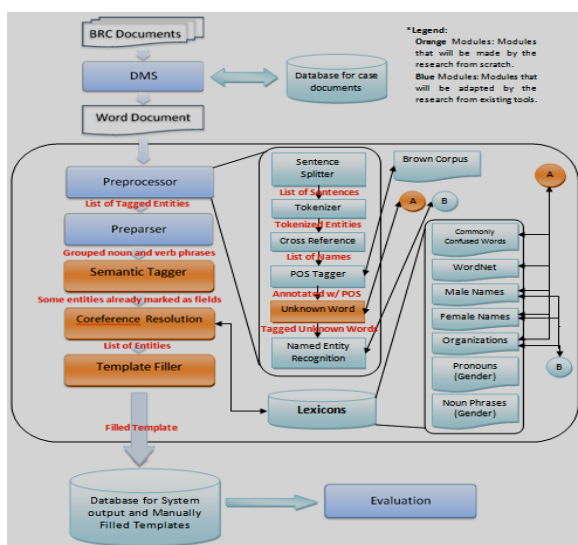


Figure 4. Information Extraction System Architecture for e-Participation

The Pre-Parser module follows pre-processing stage. This module establishes the phrases (noun and verb phrases) in the POS-tagged sentences using LingPipe (2003). Having established these phrases, the Semantic Tagger follows which is responsible for extracting candidate values for each field found in a certain template of a document. The Semantic Tagger makes use of the outputs provided by the previous modules to determine the correct candidate values. It basically goes over a certain document and finds the candidates by the use of semantic rules or patterns. The Co-Reference Resolution follows which uses the algorithm of Ruslan Mitkov (1998) for anaphora resolution. The algorithm was augmented to address cataphoric resolutions, which were present in the documents of the agency.

Finally, the Template filler follows which normalizes the Semantic Tagger entries such as

dates, values and names to standardize these entries before adding to the database.

3 Forums + Opinion Classification and Clustering = Opinion Organization

For the ground-up (or bottom-up) participation component, a web-based opinion detection and classification system, aptly named Vox Pop, was developed. It allows for the public to voice out their opinions regarding topics of discussion (created by moderators) by posting on the e-Legislation system. Vox Pop is able to detect opinions based on the input text of the respondents, annotate opinions to separate them from non-opinions, classify opinions by polarity (as shown in Figure 5) and by topic, clustered together these opinions, and present them through graphical representations of the data. The system has three main modules, namely: the opinion detection module, the opinion classification module and the clustering module.

Again, a whole discourse on managing forums and netizens (or e-citizens) as well as the processes adopted for promoting, regulating and cultivating skills of netizenship are very related and in fact, determined the configuration and business processes of the forum. Nevertheless, the focus of this section is on the technical aspect or the Language Technology used in the forum.

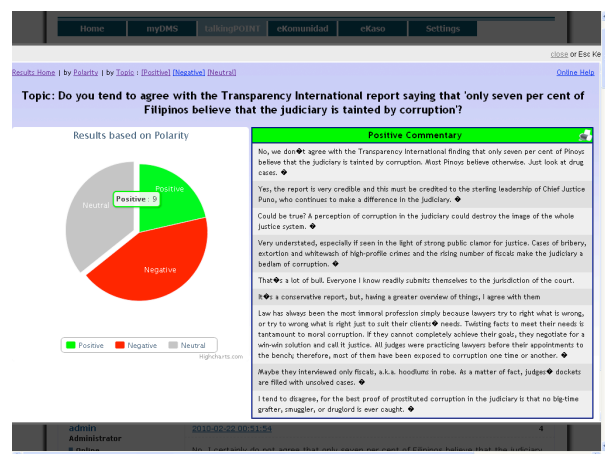


Figure 5. Opinion Polarity Report Screenshot

3.1 Detection

Commentaries are first gathered into a repository then go through the first module of the system, which is the Opinion Detection Module. The first module detects quotations and opinions present within the commentary. The heuristic goes by using signal phrases to detect quotations taken from previous posts or passages from other

sources that are present within the commentary. The presence of quotations taken from previous posts, which can be considered as opinion spamming, could duplicate commentaries within the topic thus resulting in the presence of an additional positive or negative score. This feature prevents the occurrence of these duplicate commentaries or Opinion Spam for a more accurate positive or negative result.

Sentences that are tagged as opinions are forwarded to the classification module while the sentences that are untagged or tagged as quotations are disregarded in the succeeding module.

The output of the opinion detection module is the combination of the detection of quotations and the detection of opinions. These two detection processes are needed in order for the classification process to determine which parts of the commentary are to be forwarded to the Opinion Classification Module. Lines of text that are tagged as opinions are selected to undergo the classification process while sentences that are not tagged or are tagged as quotations will not undergo the classification process.

3.2 Classification

After the Opinion Detection module tags commentaries as opinions, all opinions are then classified and tagged with their polarity. To determine the polarity of an opinion, it goes through four steps, namely: Part of Speech Tagging, Polarity Score Generation, Polarity Score Computation and Determining the Polarity of the Commentary. This module uses MontyTagger (Liu, 2003) for part-of-speech tagging and SentiWordNet (Esuli & Sebastiani, 2006) for polarity score generation.

In computing the Polarity score, there are three levels of computation, namely: Word-level, Sentence-level and Commentary-level. In the computation for the word level polarity, the Positivity and Negativity scores of all of the synsets of a particular adjective or adverb, depending on use, will be averaged in order to compute for the scores of that particular word.

After computing for the word level scores, the Positivity and Negativity scores of all adjectives and adverbs in a particular sentence will be added and then averaged in order to come up with the scores of that particular sentence. Finally, this process is repeated one more time, this time adding and averaging the scores of sentences, in order to come up with the commentary-level scores.

3.3 Clustering

After being classified by polarity, the commentaries would then be clustered by topic. Each commentary would first undergo two types of pre-processing, namely, stop words removal and stemming of words. After pre-processing the commentaries, the mean of each commentary would then be computed, and then the Euclidean distance between the commentaries and will finally be subjected to the K-Means Clustering proper.

The clustering algorithm used by the system is based on Collaborative Decision Support System (CoDeS) (Chiu et al., 2008). However, the implementation is slightly altered from CoDeS. While CoDeS accepts commentaries without any pre-processing for clustering, Vox Pop's clustering module accepts commentaries which are already classified by polarity by the classification module.

4 Prototype Results and Preliminary Findings

The e-Legislation portal integrates the Document Management System (DMS) and the Online Forums with the Information Extraction and Opinion Organization technologies, respectively (see Figure 6). Moreover, the portal provides for features that exploit the structured information coming from the two language technologies and allows users to access or view these structured data. For the Document Management System with Information Extraction, keyword searches are not limited to just the filenames since more complex database queries are available. Moreover, visual modes of reporting by exploiting the structured database from information extraction are made available. An example can be shown in Figure 7 where case activities of the Senate Committee can be visually reported thru a timeline by utilizing extracted date information from related documents in a case.

The scheme of the interaction of the DMS and the online forum, as well as the rules and regulations established in the study for governing the forums, hinges on principles of e-Democracy. These discussions, again, covers another whole set of discourse that will not be covered by this paper. Nevertheless, the same principles of e-Democracy lead to the scheme of having two forums that addresses issues of inclusivity and exclusivity (Islam, 2008) of netizens and having

set-up the forums as a self-managed system of e-Participation as envisioned by Banathy (1996).

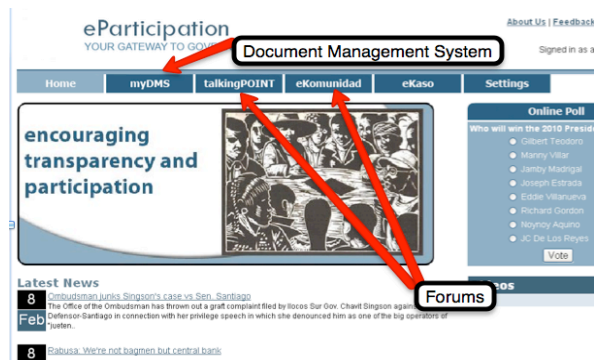


Figure 6. e-Participation Portal Integrating DMS and Online Forums



Figure 7. Timeline Report for Senate Blue Ribbon Committee Cases

The subsequent subsections will provide insights as to the evaluation and results of the performance of the two Language Technologies used for the e-Legislation Portal.

4.1 Testing Setup for DMS with Information Extraction

The study used 50 documents from the agency (Blue Ribbon Committee) for evaluation. The system supports seven kinds of documents and of those 50 documents, 5 are notice of hearing documents, 5 are agenda documents, 5 are order documents, 12 are subpoena documents, 10 are scenario documents, 8 are hearing invitation documents, and 5 are hearing highlights documents. Each type of document has their own set of fields and one of every type of document was used as basis in obtaining the rules for the various fields for the document of its type. Each notice of hearing document has the same format across every notice of hearing documents and this also applies to the other type of documents. Establishing the fields for each document type involved a series of focus-group discussions with

different stakeholders ranging from private citizens to non-government organizations (NGO's) and even to representatives from Commission (government) on Women.

In evaluating the extraction performance in terms of accuracy, the representatives from the government agency manually derived and provided the gold standard (or the answer key). The gold standard is matched with the output of the information extraction module to which a score of an absolute score of 1 or 0 is given if it's an exact match or not, respectively. A non-exact match also constitute added, removed or altered words from the correct answer.

The accuracy of the system from the training data given is 100 percent accurate, the reason for this output is because the system is constructed with the training data so it shows that the system is implemented and tested side by side with the training data. The output of the system is based on the pattern that was given to the research by the Blue Ribbon Committee. The resource person indicated that the pattern is being followed for all the documents, so as long as the pattern is being followed by the implementation of the system, then the output would be accurate.

4.2 Testing the Actual Data for Information Extraction

Unit-testing the information extraction from the training data showed a 100 percent accuracy performance. In other words, the algorithms of the sub-modules of information extraction were finely tuned for the doing things correctly. After testing the system with the training data, a new set of 50 documents was evaluated. The output of the system is mostly the same with only a small set of differences due to the mismatch of format of values from the training versus the actual data. In other words, there are certain parts in the new test data that are not found in the training or initial test data. One attribute to the disparity is on the irregularity of the document formatting. Another would be the addition of fields and entities manually extracted by the resource people creating the gold standards. **Table 1** shows the result of the actual data and only two types of documents have inaccuracies. Overall, averaging the performance of the system, it shows a 95.42% accuracy.

Table 1. Actual Data Test Results

Number of Docs.	Type of Docs.	Accuracy
4	Notice of Hearing	100%
6	Agenda	100%
5	Order	100%
17	Subpoena	85%
7	Scenario	100%
6	Hearing Invitation	83%
5	Hearing Highlight	100%

4.3 Testing Setup for Forums and Opinion Organization

The corpus was built from commentaries obtained from the website of the Philippine Star Inbox World (www.philstar.com). It contains 1002 commentaries from 22 topics. A linguist evaluator was tapped and did six sets of evaluations, four of which were for the classification module; one was for the detection module, while the last set was for the clustering module.

4.4 Detection Results

In order to check whether detected opinions are correctly annotated by the system, the same set of commentaries used to evaluate the classification module was fed into the opinion detection module. All two hundred commentaries were split into sentences and were determined whether they are opinionated or not. One hundred opinionated and another one hundred non-opinionated sentences, as determined by the system, were then randomly selected from the output to be evaluated by the linguist whether they are opinionated or not. All in all, two hundred (200) sentences from one hundred and one (101) commentaries were selected to be evaluated by the linguist. Of the two hundred sentences evaluated by the linguist, one hundred commentaries or **50%** matched with the detection done with the system.

An observation regarding this matter is that even though there are an equal number of ‘opinionated’ and ‘non-opinionated’ commentaries as tagged by the system, the evaluators tagged a much greater number of commentaries as ‘opinionated’. This may be due to the evaluators having a pre-conceived notation that most of the commentaries are opinionated because of the knowledge that the commentaries come from an opinionated source. However, the system does not know this and instead, bases judgment on the opinion markers present in a commentary.

Another observation in the opinion detection module is that the signal phrases that are used to compare can occur in a given sentence even without a following passage that is enclosed in quotation marks. For example, the signal phrase ‘goes’ can be used in a sentence as a signal phrase in (*There’s a saying that goes, "The road to hell is paved with good intentions."*) and as a verb in (*There goes the alarm*). Majority of the signal phrases are verbs and these verbs may be confused as being verbs without the support of a passage enclosed in quotation marks.

Another observation to the system is that there are other signal phrases that are not included in the overall list of signal phrases used by the system. These signal phrases follow the same structure in quoting passages and sayings but if the signal phrase is not present in the overall list of signal phrases used by the system, the system may not consider it as a quotation. There may also be opinion markers that are not included in the overall list of opinion markers used by the system. These opinionated sentences with markers not found in the list may not be considered by the system as an opinion.

Finally, it was observed that the comparison of opinion markers and signal phrases cannot be accomplished when comparing them with the words in the sentence. This is because of the fact that the lists do not only contain a single word but there are also phrases contained within the list. Examples would be “points out” and “according to” for opinion markers and signal phrases respectively. These observations found in the system may have contributed to the accuracy rate of the opinion detection module.

4.5 Classification Results

In order to check whether classified opinions are correctly annotated by the system, two hundred commentaries were chosen from the corpus and were subjected to polarity annotation by the system. The linguist was then requested to identify the polarity of each commentary. Of the two hundred commentaries evaluated by the linguist, one hundred one commentaries or **50.5%** matched with the classification done with the system.

Three sources of errors were found. The first error is the failure to process double-negative phrases. The system gets the individual scores of the words in a commentary then adds them afterwards. An example would be the statement *“Another Aquino to dominate the Philippine as a leader has my yes. I hope Noynoy does not fail*

us.” This statement is evaluated by the linguist as positive, as the commentary is a statement of support for Noynoy Aquino. The word ‘Yes’ alone tells the human brain to evaluate the statement as positive. However, the formula used fails when faced with double negatives, or negative words placed next to the word ‘not’. The example sentence was marked as negative by the system because the words ‘not’ and ‘fail’, which are both negative in nature, were evaluated separately from each other by the system. This is why the negativity score of the statement increased, instead of the positivity score increasing if it were processed as one statement ‘not fail’.

The second error is the presence of high polarity words. Since the range of the scores of the words is normalized from 0 to 1, it is possible for several words of a particular polarity to overpower a word of the opposite polarity. An example of this would be the statement “*I believe he would make a good president of our country if given a chance, but he should be a good senator first.*” This statement is evaluated by the linguist as negative, as the ‘but’ part of the commentary is not in support of Noynoy Aquino. However, it was marked as positive by the system. Although the word ‘but’ has a high negativity score, the positivity score of the words ‘believe’ and ‘good’, which appeared twice, overpowered the score of the word ‘but’ because there are more words present which have high positivity scores.

The third error occurs when adjectives and adverbs are absent in a sentence. Adjectives and adverbs contain the sentiment of a statement. That is why in SentiWordNet, adjectives and adverbs have non-zero positivity and negativity scores. However, if a statement does not contain adjectives and adverbs, the positivity and negativity scores of these statements are both zero, leading the system to classify them as neutral. An example would be the statement “*A hundred percent yes.*” This statement is evaluated by the linguist as positive, as the word “Yes” is enough to tell that the sentiment is positive. However, it was marked by the system as neutral because the words ‘a’, ‘percent’ and ‘yes’ are nouns, while the word ‘hundred’ is an adjective which has both zero positivity and negativity scores.

4.6 Clustering

In order to check whether the clusters produced by the system are correct, two topics containing eighty one (81) commentaries were chosen to be clustered by the system and evaluated by the linguist afterwards. In generating the clusters, the

commentaries in each topic were first segregated into three clusters, produced by the opinion classification module of the system. Afterwards, each polarity-segregated cluster was further segregated into three smaller clusters. Thus, all in all, eighteen clusters were produced by the system for the evaluation of the linguist. The clusters generated by the system were analyzed by the linguist whether 1) the commentaries in each cluster are related with each other, and 2) why some commentaries are singled out into single clusters. Of these eighteen clusters, thirteen contain multiple commentaries, while the remaining five contain only single commentaries.

In the first topic, three unrelated clusters were deemed as such because the views in them are complex, vary with each other and some go beyond the intended topic. In the second topic, the three unrelated clusters were deemed as such because their views vary from each other. Another unrelated cluster contained an opinion and a declaration of support. These commentaries are grouped together because of the similar words that are found within them, such as ‘Filipino’ and ‘honesty’ in the first topic. However, the clustering algorithm does not take into account synonyms or words that are similar in context, resulting to some clusters being mismatched. All in all, the linguist evaluation shows a **53.85%** accuracy of the clustering module.

In the process of clustering, five commentaries were not clustered and instead, were isolated from the other commentaries. Of the five clusters containing only one commentary each, two of them were evaluated by the linguist being isolated because they sound like factual statements. On the other hand, the other two were evaluated by the linguist as being isolated because they contain alternate reasons on why they agree or disagree with the topic. Finally, the last cluster containing only one commentary was probably isolated because “*it cites very specific instances*”, as the linguist points out.

These commentaries were isolated probably because the default number of clusters (three) is not the optimum number of clusters for the set of commentaries. An example would be the positive clusters under the second topic. When the number of clusters was set to three, one commentary was isolated while the other two clusters contained three and thirteen commentaries respectively. However, when the number of clusters was set to two, no more commentaries were isolated and two clusters containing multiple commentaries are formed.

5 Conclusions and Recommendations

Overall, the document management system was designed for the Blue Ribbon Oversight Office Management (BROOM) after constant requirements gathering, consultation and collaboration with BROOM. The Information Extraction module was also designed implemented and evaluated in the same manner and thus garnered very high extraction accuracy.

The system currently isn't robust enough to handle noise brought about by poor conversion of hardcopies to electronic versions (e.g. Optical Character Recognition). Moreover, the algorithms are highly dependent on the regularity of the documents and would perform differently if documents don't conform to regular formats.

It is recommended that improving the Information Extraction module entail the following:

1. Image recognition algorithms that allows for capturing and extracting signatures from senators and annotation information as extracting these data allows for interesting research and data mining value;
2. Improvement of semantic tagger to handle new templates without changing the code but instead process new documents from templates based on formal specifications; and
3. Include Filipino processing in extracting texts as transcript-type of documents would normally contain a mixture and code switching of English and Filipino languages.

For the opinion organization, the study focused on developing a system that uses text processing techniques in organizing the sentiments of public commentary.

The opinion detection module includes the detection of quotations and opinions given input commentaries to which opinion classification is affected. Quotation detection prevents quotations from being redundantly classified, thus providing more accurate results for classification.

The opinion classification module included part-of speech tagging, polarity score generation via SentiWordNet and word, sentence and commentary-level score computations. It was uncovered that part of speech tagging is important as adjectives and adverbs really do have the linguistic basis in classifying commentaries by sentiment. However, it was also shown that SentiWordNet should not be the sole tool used in dealing with polarity, as it only outputs the score of each word, and it does not consider more

complex factors such as double negatives and idioms.

The clustering module includes stop words removal, stemming and the use of the K-Means clustering algorithm. Stop words removal and stemming are necessary in clustering as they filter commentaries, preventing non-relevant words such as prepositions, articles and pronouns from being used as the basis for clustering. However, having a fixed number of clusters, generated by the K-Means clustering algorithm, which is three in this case, is not the most optimal solution for all cases. If there are only few commentaries to be clustered, setting the number of clusters to a smaller number such as two might be more optimal. Conversely, three clusters might not be sufficient for a larger dataset, such as the ones containing thousands of commentaries in them.

All of these issues attributed to the dismal 50% overall accuracy performance of the opinion organization and classification and data clustering. Nevertheless, the different presentations and structured reporting of commentaries and opinion facilitated by the automated detection, classification and clustering still provide a way for structuring information that can facilitate policy making or legislation.

It is recommended that improving the automated opinion organization entail the following:

1. As with the Information Extraction Module, include Filipino processing as texts in comments and opinions also include a mixture and code switching of English and Filipino languages;
2. Utilize SentiWordnet version 3.0 which increased by 20% in accuracy versus Version 1.0 (Baccianella, et al. 2010), as the current implementation involves version 1.0; and
3. Investigate machine learning on top of relying on lexical and rule-based resource such as SentiWordnet to allow for flexibility and robustness of system;

For both major modules addressing the top-down and bottom-up information, linguistic resources and algorithms are still currently being improved. But more importantly, the Blue Ribbon Oversight Office Management (BROOM) of the Philippine Senate is now steadily migrating to a new business process, creating the possibility of allowing the office to open its documents to the public (towards transparency and good governance) more expeditiously, and allowing feedback from citizenry (towards participation)

as the office is currently moving to actual adoption of the eLegislation Portal.

Acknowledgments

The e-Legislation Portal was greatly facilitated by the Blue Ribbon Oversight Office Management (BROOM) of the Philippine Senate headed by Atty. Rodolfo Quimbo, Director General who served as the “champion” in government for this endeavour. Successful business process modeling of the said office due to solid support and constant interaction led to a very functional and seamless integration of office automation and portal for pushing information to general public. Moreover, constant interaction and support for quality evaluation led to a robust and accurate information extraction performance. Dr. Shirley Lua, linguist and faculty member of the Literature Department of De La Salle University, did the daunting task of evaluating the opinion organization’s performance.

The whole project is under the PanEGov project managed by IdeaCorp headed by the executive director, Dr. Emmanuel C. Lallana and funded by IDRC, Canada.

References

- Ann Macintosh. 2007. e-Democracy and e-Participation Research in Europe. In Chen, et al (eds.) *Digital Government: E-Government Research, Case Studies, and Implementation*. Springer.
- Hobbs, J. R. 1993. The Generic Information Extraction System. In MUC5 '93: Proceedings of the 5th Conference on Message Understanding (pp. 87-91). Morristown, NJ, USA: Association for Computational Linguistics. Available from <http://dx.doi.org/10.3115/1072017.1072029>
- Schwartz, A. and Hearst, M. 2003. A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. *Pacific Symposium on Biocomputing* 8:451-462. Available: <http://helix-web.stanford.edu/psb03/schwartz.pdf>
- LingPipe: a suite of Java libraries for the linguistic analysis of human language. 2003-2007. Accessed March 2010. <http://ir.exp.sis.pitt.edu/ne/lingpipe-2.4.0/index.html>
- ANNIE: POS Tagger (Plugins for GATE). 1995-2011. Accessed March 2010. <http://gate.ac.uk/gate/doc/plugins.html>
- Mitkov, R. 1998. Robust Pronoun Resolution with Limited Knowledge. In *Acl-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics* (pp. 869–875). Morristown, NJ, USA: Association for Computational Linguistics. Available from <http://dx.doi.org/10.3115/980691.980712>
- Esuli, A., Sebastiani, F. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In Proceedings of the 5th Conference on Language Resources and Evaluation (pp. 417-422).
- Liu, Hugo. 2003. *Monty Tagger: Commonsense-Informed Part-of-Speech Tagging*. Available online: <http://web.media.mit.edu/~hugo/montytagger/>
- Chiu, C., Dy, J., Lee, P., & Liu, M. 2008. *Collaborative Decision Support System* [Thesis]. Manila, Philippines: College of Computer Studies, De La Salle University.
- Islam, M.S. 2008. *Towards a Sustainable e-Participation Implementation Model*. European Journal of e-Practice. Available online: <http://www.epractice.eu/files/5.3.pdf>
- Banathy, Bela H. 1996. *Designing Social Systems in a Changing World*. New York, Plenum.
- Baccianella S., Esuli A. & Sebastiani F. 2010. *SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining*. In: LREC 2010 - Seventh conference on International Language Resources and Evaluation (Valletta, Malta, 18-22 maggio 2010). Proceedings, pp. 2200 - 2204. ELRA, 2010.

Author Index

- , Chungku, 148
- Abate, Solomon Teferra, 50, 128
Adegbola, Tunde, 7
Adly, Noha, 118
Al Ghamdi, Mansour, 162
Alam, Firoj, 87, 154
Alansary, Sameh, 118
Anberbir, Tadesse, 68
Aye, Dr.Khin, 27
- Barnard, Etienne, 81
Besacier, Laurent, 50, 128
Borra, Allan, 196
Buckley, Jim, 169
- Chagnaa, Altangerel, 56
Chege, Kamau, 112
Cheng, Charibeth, 196
Chhoeden, Dechen, 62, 148
Chhoejey, Pema, 62
Choejey, Pema, 34
Chotimongkol, Ananlada, 148
Chowdhury, Shammur Absar, 87
Chungku, Chungku, 34
- De Pauw, Guy, 44, 106, 112
Dilawari, Aniqā, 11
Dong Yoon, Kim, 68
Durrani, Qaiser, 134
- Gasser, Michael, 68, 94
Gelas, Hadrien, 128
Gizaw, Solomon, 169
- Habib, S.M. Murtoza, 87, 154
Herath, Dulip Lakmal, 39
Hoogeveen, Doris, 44
Htay Hlaing, Tin, 1
Hussain, Sarmad, 11, 187
- Ijaz, Madiha, 134
- Jaimaa, Purev, 56
Jayawardhane, Tissa, 39
- Khan, Mumit, 87, 154
- Liyanage, Chamila, 39, 176
Liyanapathirana, Jeevanthi Uthpala, 182
- Mikami, Yoshiki, 1
Mutiga, Jayne, 112
- Nadungodage, Thilini, 141
Nagi, Magdy, 118
Ng'ang'a, Wanjiku, 112
Nganga, Wanjiku, 100
Nzioka Kituku, Benson, 106
- ODEJOBI, Odetunji Ajadi, 74
- Pellegrino, François, 128
Pemo, Dawa, 62
Pushpananda, Randil, 176
- Rabgay, Jurmey, 34
Roxas, Rachel, 196
Rugchatjaroen, Anocha, 148
- Sarfraz, Huda, 11
Shams, Sana, 187
Sharma Grover, Aditi, 81
Sherpa, Uden, 62
Sultana, Rabia, 87
- Tachbelie, Martha Yifiru, 50
Takara, Tomio, 68
Tawileh, Anas, 162
Than, Dr.Myint Myit, 27
Than, Moh Moh, 27
Thangthai, Ausdang, 148
- Udalamatta, Namal, 39, 176
- Vanthanavong, Sisouvanh, 21
- W. Wagacha, Peter, 106, 112
Weerasinghe, Ruvan, 39, 141, 176, 182
Welgama, Viraj, 39
Win, May Thu, 27
Win, Moet Moet, 27
Wutiwiwatchai, Chai, 148