Dzongkha Word Segmentation

Sithar Norbu, Pema Choejey, Tenzin Dendup
Research Division
Department of Information Technology & Telecom
{snorbu, pchoejay, tdendup}@dit.gov.bt

Sarmad Hussain, Ahmed Mauz
Center for Research in Urdu Language Processing
National University of Computer & Emerging Sciences
{sarmad.hussain, ahmed.mauz}@nu.edu.pk

Abstract

Dzongkha, the national language of Bhutan, is continuous in written form and it fails to mark the word boundary. Dzongkha word segmentation is one of the fundamental problems and a prerequisite that needs to be solved before more advanced Dzongkha text processing and other natural language processing tools can be developed. This paper presents our initial attempt at segmenting Dzongkha sentences into words. The paper describes the implementation of Maximal Matching (Dictionary based Approach) followed by bigram techniques (Non-dictionary based Approach) in segmenting the Dzongkha scripts. Although the used techniques are basic and naive, it provides a baseline of the Dzongkha word segmentation task. Preliminary experimental results show percentage of segmentation accuracy. However, the segmentation accuracy is dependent on the type of document domain and size and quality of the lexicon and the corpus. Some of the related issues for future directions are also discussed.

Keywords: Dzongkha script, word segmentation, maximal matching, bigram technique, smoothing technique.

1 Introduction

Segmentation of a sentence into word is one of the necessary preprocessing tasks and is essential in the analysis of natural language processing. This is because word is both syntactically and semantically, the fundamental unit for analyzing language structure. Like in any other language processing task, Dzongkha word segmentation is also viewed as one of the fundamental and foremost steps in Dzongkha related language processing tasks.

The most challenging features of Dzongkha script is the lack of word boundary separation between the words[1]. So, in order to do the further linguistic and natural language processing tasks, the scripts should be transformed into a chain of words. Therefore, segmenting a word is an essential role in Natural Language Processing. Like Chinese, Japanese and Korean (CJK) languages, Dzongkha script being written continuously without any word delimiter causes a major problem in natural language processing tasks. But, in case of CJK, Thai, and Vietnamese languages, many solutions have been published before. For Dzongkha, this is the first ever word segmentation solution to be documented.

In this paper, we describe the Dzongkha word segmentation, which is performed firstly using the Dictionary based approach where the principle of maximal matching algorithm is applied to the input text. Here, given the collection of lexicon, the maximal matching algorithm selects the segmentation that yields the minimum number of words token from all possible segmentations of the input sentence. Then, it uses non-dictionary based approach where bigram technique is applied. The probabilistic model of a word sequence is

[1] http://www.learntibetan.net/grammar/sentence.htm

Proceedings of the 8th Workshop on Asian Language Resources, pages 94–101,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

studied using the Maximum Likelihood Estimation (MLE). The approach using the MLE has an obvious disadvantage because of the unavoidably limited size of the training corpora (Nuges, 2006). To this problem of data sparseness, the idea of Katz back-off model with Good-Turing smoothing technique is applied.

2 Dzongkha Script

Dzongkha language is the official and national language of Bhutan. It is spoken as the first language by approximately 130,000 people and as the second language by about 470,000 people (Van Driem and Tshering, 1998).

Dzongkha is very much related to Sino-Tibetan language which is a member of Tibeto-Burmese language family. It is an alphabetic language, with phonetic characteristics that mirror those of Sanskrit. Like many of the alphabets of India and South East Asia, the Bhutanese script called Dzongkha script is also a syllabic[2]. A syllable can contain as little as one character or as many as six characters. And a word can be of one syllable, two syllable or multiple syllables. In the written form, Dzongkha script contains a dot, called Tsheg (་) that serve as syllable and phrase delimiter, but words are not separated at all.

For example,

Dzongkha	Transliteration	English	Syllables
དམརཔོ	dmarpo	red	Single-syllabled
སློབ་དཔོན	slop-pon	Teacher	Two-syllabled
འཇམ་ཏོག་ཏོ	hjam-tog-to	easy	Three-syllabled
འར་རི་ཧུར་རི	har-ri-hur-ri	crowdedness /confusion	Four-syllabled

Table 1: Different syllabled Dzongkha scripts.

The sentence is terminated with a vertical stroke called Shad (།). This Shad acts as a full_stop. The frequent appearance of whitespace in the Dzongkha sentence serves as a phrase boundary or comma, and is a faithful representation of speech: after all in speech, we pause not between words, but either after certain phrases or at the end of sentence.

The sample dzongkha sentence reads as follows:

རྫོང་ཁ་གོང་འཕེལ་ལྷན་ཚོགས་འདི་ འབྲུག་རྒྱལ་ཁབ་ནང་ གཞུང་གི་ཁ་ཐུག་ལས་ འབྲུག་གི་རྒྱལ་ཡོངས་སྐད་ཡིག་ རྫོང་ཁའི་སྲིད་བྱུས་རྩམ་མི་དང་ རྫོང་ཁའི་མཐར་ཐུག་གི་དབང་འཛིན་པ་ རང་དབང་རང་སྐྱོང་གི་འདུས་ཚོགས་ མཐོ་ཤོས་ཅིག་ཨིན། འདུས་ཚོགས་འདི་ འབྲུག་རྒྱལ་བཞི་པ་མི་དབང་མངའ་བདག་རིན་པོ་ཆེ་ དཔལ་འཇིགས་མེད་སེང་གེ་དབང་ཕྱུག་མཆོག་གི་ཐུགས་དགོངས་དང་འཁྲིལ་ཏེ་ སྤྱི་ལོ་ ༡༩༨༦ ལུ་གཞི་བཙུགས་གནང་གནངམ་ཨིན།

(English Translation of example text)

[The Dzongkha Development Commission is the leading institute in the country for the advancement of Dzongkha, the national language of Bhutan. It is an independent organization established by the Fourth King of Bhutan, His Majesty the King Jigme Singye Wangchuck, in 1986.]

3 Materials and Methods

Since, our language has no word boundary delimiter, the major resource for Dzongkha word segmentation is a collection of lexicon (dictionary). For such languages, dictionaries are needed to segment the running texts. Therefore, the coverage of a dictionary plays a significant role in the accuracy of word segmentation (Pong and Robert, 1994).

The dictionary that we used contains 23,333 word lists/lexicons. The lexicons were collected from “Dzongkha Dictionary”, 2nd Edition, Published by Dzongkha Development Authority, Ministry of Education, 2005, (ddc@druknet.bt). The manually segmented text corpus containing 41,739 tokens are also used for the method. The text corpora were collected from different sources like newspaper articles, dictionaries, printed books, etc. and belong to domains such as World Affairs, Social Sciences, Arts, Literatures, Adventures, Culture and History. Some texts like poetry and songs were added manually.

[2] http://omniglot.com/writing/tibetan.htm

Table below gives the glimpse of textual domains contained in the text corpora used for the method (Chungku et al., 2010).

Domain	Sub domain	(%)
World Affairs	Bilateral relations	12%
Social Science	Political Science	2%
Arts	Poetry/Songs/Ballad	9%
Literatures	Essays/Letters/Dictionary	72%
Adventures	Travel Adventures	1%
Culture	Culture Heritage/Tradition	2%
History	Myths/Architecture	2%

Table 2: Textual domain contained in Corpus

Figure 1 below shows the Dzongkha Word Segmentation Process.

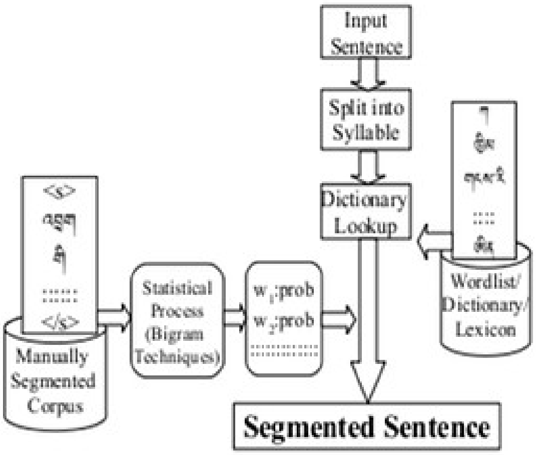
Input Sentence
Split into Syllable
Dictionary Lookup
Wordlist/ Dictionary/ Lexicon
<s>
</s>
Manually Segmented Corpus
Statistical Process (Bigram Techniques)
w1:prob
w2:prob
Segmented Sentence

Figure 1: Dzongkha Word Segmentation Process.

Dzongkha word segmentation implements a principle of maximal matching algorithm followed by statistical (bigram) method. It uses a word list/lexicon at first to segment the raw input sentences. It then uses MLE principles to estimate the bigram probabilities for each segmented words. All possible segmentation of an input sentence by Maximal Matching are then re-ranked and picked the mostly likely segmentation from the set of possible segmentations using a statistical approach (bigram technique). This is to decide the best possible segmentation among all the words (Huor et al., 2007) generated by the maximal matching algorithm. These mechanisms are described in the following

3.1 Maximal Matching Algorithm

The basic idea of Maximal matching algorithm is, it first generates all possible segmentations for an input sentence and then selects the segmentation that contains the minimum number of word tokens. It uses dictionary lookup.

We used the following steps to segment the given input sentence.

1. Read the input of string text. If an input line contains more than one sentence, a sentence separator is applied to break the line into individual sentences.
2. Split input string of text by Tsheg(་) into syllables.
3. Taking the next syllables, generate all possible strings
4. If the number of string is greater than *n* for some value *n*[3]
   - Look up the series of string in the dictionary to find matches, and assign some weight-age[4] accordingly.
   - Sort the string on the given weight-age
   - Delete (number of strings – *n*) low count strings.
5. Repeat from Step 2 until all syllables are processed.

The above mentioned steps produced all possible segmented words from the given input sentence based on the provided lexicon. Thus, the overall accuracy and performance depends on the coverage of lexicon (Pong and Robert, 1994).

[3] The greater the value of n, the better the chances of selecting the sentence with the fewest words from the possible segmentation.

[4] If the possible string is found in the dictionary entries, the number of syllable in the string is counted. Then, the weight-age for the string is calculated as (number of syllable)^2 else it carries the weight-age 0

3.2 Bigram Method

(a) **Maximum Likelihood Estimation**[5]

In the bigram method, we make the approximation that the probability of a word depends on identifying the immediately preceding word. That is, we calculate the probability of next word given the previous word, as follows:

$$P\left(w_1^n\right) = \Pi_{i=1}^n P\left(w_i / w_{i-1}\right)$$

where

- $$P\left(w_i / w_{i-1}\right) = \frac{count\left(w_{i-1} w_i\right)}{count\left(w_{i-1}\right)}$$

where

- $count\left(w_{i-1} w_i\right)$ is a total occurrence of a word sequence $w_{i-1} w_i$ in the corpus, and
- $count\left(w_{i-1}\right)$ is a total occurrence of a word w_{i-1} in the corpus.

To make $P\left(w_i / w_{i-1}\right)$ meaningful for $i=1$, we use the distinguished token <s> at the beginning of the sentence; that is, we pretend w_0 = <s>. In addition, to make the sum of the probabilities of all strings equal 1, it is necessary to place a distinguished token </s> at the end of the sentence.

One of the key problems with the MLE is insufficient data. That is, because of the unavoidably limited size of the training corpus, vast majority of the word are uncommon and some of the bigrams may not occur at all in the corpus, leading to zero probabilities.

Therefore, following smoothing techniques were used to count the probabilities of unseen bigram.

(b) **Smoothing Bigram Probabilistic**

The above problem of data sparseness underestimates the probability of some of the sentences that are in the test set. The smoothing technique helps to prevent errors by making the probabilities more uniform. Smoothing is the process of flattening a probability distribution implied by a language model so that all reasonable word sequences can occur with some probability. This often involves adjusting zero probabilities upward and high probabilities downwards. This way, smoothing technique not only helps prevent zero probabilities but the overall accuracy of the model are also significantly improved (Chen and Goodman, 1998).

In Dzongkha word segmentation, Katz back-off model based on Good-Turing smoothing principle is applied to handle the issue of data sparseness. The basic idea of Katz back-off model is to use the frequency of n-grams and if no n-grams are available, to back off to *(n-1)* grams, and then to *(n-2)* grams and so on (Chen and Goodman, 1998).

The summarized procedure of Katz smoothing technique is given by the following algorithm:[6]

$$P_{katz}\left(w_i | w_{i-1}\right) = \left\{ \begin{array}{c} C\left(w_{i-1}/w_i\right) \;\; if r>k \\ d_r C\left(w_{i-1}/w_i\right) \;\; if k \geq r>0 \\ \alpha\left(w_{i-1}\right) P\left(w_i\right) \;\; if r=0 \end{array} \right\}$$

where

- r is the frequency of bigram counts
- k is taken for some value in the range of 5 to 10, other counts are not re-estimated.
- $$d_r = \frac{\frac{r^*}{r} - (k+1)\frac{n_{K+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}$$
- $$\alpha\left(w_{i-1}\right) = \frac{1 - \sum_{w_i : r>0} P_{Katz}\left(w_i | w_{i-1}\right)}{1 - \sum_{w_i : r>0} P_{Katz}\left(w_i\right)}$$

With the above equations, bigrams with non-zero count *r* are discounted according to the

[5] P.M, Nugues. An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with Special Consideration of English, French, and German (Cognitive Technologies) (95 – 104).

[6] X. Huang, A. Acero, H.-W.Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, (Prentice-Hall Inc., New Jersey 07458, 2001), 559 - 561.

discount ratio $d_r = \frac{r^*}{r}$ i.e., the count subtracted from the non-zero count are redistributed among the zero count bigrams according to the next lower-order distribution, the unigram model.

4 Evaluations and Results

Subjective evaluation has been performed by comparing the experimental results with the manually segmented tokens. The method was evaluated using different sets of test documents from various domains consisting of 714 manually segmented words. Table 3 summarizes the evaluation results.

Document text	Correct Detect (Correctly segmented tokens / total no. of words)	Accuracy
Astrology.txt	102/116	87.9%
dzo_linux.txt	85/93	91.4%
movie_awards.txt	76/84	90.5%
News.txt	78/85	91.8%
Notice.txt	83/92	90.2%
Religious.txt	63/73	89.0%
Song.txt	57/60	95.0%
Tradition.txt	109/111	98.2%
Total	**653/714**	**91.5%**

Table 3: Evaluation Results

Accuracy in %age are measured as:

$$\text{Accuracy(\%)} = \frac{N}{T} * 100$$

where

- N is the number of correctly segmented tokens
- T is the total number of manually segmented tokens/ Total number of words.

We have taken the extract of different test data hoping it may contain fair amount of general terms, technical terms and common nouns. The manually segmented corpus containing 41,739 tokens are used for the method.

In the sample comparison below, the symbol (་) does not make the segmentation unit's mark, but (།) takes the segmentation unit's mark, despite its actual mark for comma or full_stop. The whitespace in the sentence are phrase boundary or comma, and is a faithful representation of speech where we pause not between words, but either after certain phrases or at the end of sentence.

Consider the sample input sentence:
རྫོང་ཁ་ལི་ནགསི་འདི་ རྫོང་ཁ་གློག་རིག་ནང་བཙུགས་ནིའི་དོན་ལུ་ རྒྱབ་
སྐྱོར་ཧྲིལ་བུ་གཅིག་ཁར་བསྡོམས་མི་ རང་དབང་ལི་ནགསི་ བཀོལ་སྤྱོད་
རིམ་ལུགས་འདི་གི་ ཉེ་གནས་སྦྲེ་མཐུན་འགྱུར་བཟོ་ཡོད་པའི་ཐོན་རིམ་
ཅིག་ཨིན། དེ་གིས་ ཆ་ཚང་སྦྲེ་སྐད་བསྒྱུར་འབད་ཡོད་པའི་ལག་ལེན་པའི་
ངོས་འདྲ་བ་ཚུ་སྟོནམ་ཨིན།

Manually segmented sentence of the sample input sentence:
རྫོང་ཁ།ལི་ནགསི།འདི། རྫོང་ཁ།གློག་རིག།ནང།བཙུགས།ནིའི།དོན་ལུ།
རྒྱབ་སྐྱོར།ཧྲིལ་བུ།གཅིག།ཁར།བསྡོམས།མི། རང་དབང།ལི་ནགསི། བཀོལ་
སྤྱོད།རིམ་ལུགས།འདི།གི། ཉེ་གནས།སྦྲེ།མཐུན་འགྱུར།བཟོ།ཡོད།པའི།
ཐོན།རིམ།ཅིག།ཨིན། དེ་གིས། ཆ་ཚང།སྦྲེ།སྐད་བསྒྱུར།འབད།ཡོད།པའི།
ལག་ལེན།པའི།ངོས།འདྲ་བ།ཚུ།སྟོནམ།ཨིན།

Using maximal matching algorithm:
རྫོང་ཁ། ལི། ནགསི། འདི། རྫོང་ཁ། གློག་རིག། ནང། བཙུགས།
ནིའི། དོན། ལུ། རྒྱབ་སྐྱོར། ཧྲིལ། བུ། གཅིག། ཁར། བསྡོམས།
མི། རང་དབང། ལི། ནགསི། བཀོལ་སྤྱོད། རིམ་ལུགས། འདི་གི།
ཉེ་གནས། སྦྲེ། མཐུན་འགྱུར། བཟོ། ཡོད། པའི། ཐོན། རིམ།
ཅིག་ཨིན། དེ། གིས། ཆ་ཚང། སྦྲེ། སྐད་བསྒྱུར། འབད། ཡོད།
པའི། ལག་ལེན། པའི། ངོས། འདྲ། བ། ཚུ། སྟོནམ། ཨིན།

System segmented version of the sample input sentence: Underlined text shows the incorrect segmentation.
རྫོང་ཁ། *ལི་ནགསི་འདི།* རྫོང་ཁ།གློག་རིག།ནང།བཙུགས། *ནིའི་དོན་ལུ།*
རྒྱབ་སྐྱོར།ཧྲིལ་བུ།གཅིག།ཁར།བསྡོམས།མི། རང་དབང།ལི་ནགསི།བཀོལ་
སྤྱོད།རིམ་ལུགས།འདི།གི། ཉེ་གནས།སྦྲེ།མཐུན་འགྱུར།བཟོ།ཡོད།པའི།

ཐོན|རིམ|ཅིག|ཨིན། དེ|གིས|ཆ་ཚད|སྦེ|སྐད་བསྒྱུར|འབད|ཡོད|པའི| ལག་ལེན|པའི|ངོས|འདྲ་བ|ཀྱུ|སྟོནམ། ཨིན།

5 Discussions

During the process of word segmentation, it is understood that the maximal matching algorithm is simply effective and can produce accurate segmentation only if all the words are present in the lexicon. But since not all the word entry can be found in lexicon database in real application, the performance of word segmentation degrades when it encounters words that are not in the lexicon (Chiang et al., 1992).

Following are the significant problems with the dictionary-based maximal matching method because of the coverage of lexicon (Emerson, 2000):

- incomplete and inconsistency of the lexicon database
- absence of technical domains in the lexicon
- transliterated foreign names
- some of the common nouns not included in the lexicon
- lexicon/word lists do not contains genitive endings པའི (expresses the genitive relationship as a quality or characteristic of the second element, for example, དབུལ་པའི་བུ 'son of a pauper') and འི (first singular possessive, for example, ངེའི་བུམོ which actually is ངེ་གི་བུམོ 'my daughter') that indicates possession or a part-to-whole relationship, like English 'of'.

A Dzongkha sentence like:
འདི་རྫོང་ཁ་གི་ ཞིབ་འཚོལ་ཡིག་ཆ་ ཨིན།
may include the following ambiguous possible segmentation based on simple dictionary lookup:

1.འདི|རྫོང་ཁ|གི|ཞིབ་འཚོལ|ཡིག་ཆ|ཨིན
this | Dzongkha | of | research | written document | is

2.འདི|རྫོང་ཁ|གི|ཞིབ|འཚོལ|ཡིག་ཆ|ཨིན
this | Dzongkha | of | arrange together | search/ expose | written document | is

3.འདི|རྫོང|ཁ|གི|ཞིབ་འཚོལ|ཡིག་ཆ|ཨིན
this | fortress | mouth/surface | of | research | written document | is

These problems of ambiguous word divisions, unknown proper names, are lessened and solved partially when it is re-ranked using the bigram techniques. Still the solution to the following issues needs to be discussed in the future. Although the texts were collected from widest range of domains possible, the lack of available electronic resources of informative text adds to the following issues:

- small number of corpus were not very impressive for the method
- ambiguity and inconsistent of manual segmentation of a token in the corpus resulting in incompatibility and sometimes in conflict.

Ambiguity and inconsistency occurs because of difficulties in identifying a word. Since the manual segmentation of corpus entry was carried out by humans rather than computer, such humans have to be well skilled in identifying or understanding what a word is.

The problem with the Dzongkha scripts that also hampers the accuracy of dzongkha word segmentation includes the issues such as ambiguous use of *Tsheg* (་) in different documents. There are two different types of *Tsheg:* Unicode 0F0B (་) called *Tibetan mark inter syllabic tsheg* is a normal *tsheg* that provides a break opportunity. Unicode 0F0C (༌) called *Tibetan Mark Delimiter Tsheg Bstar* is a non-breaking *tsheg* and it inhibits line breaking.
For example,
input sentence with Tsheg 0F0B:
སངས་རྒྱས་དང་ཚེ་རིང་གཉིས་ བརྡ་དོན་དང་འཕྲུལ་རིག་ནང་ ལཱ་འབད་དོ་ཡོདཔ་ཨིན་པས།
achieves 100% segmentation as follow:

སངས་རྒྱས། དང། ཚེ་རིང། གཉིས། བརྡ་དོན། དང། འཕྲུལ། རིག། ནང།
ལཱ། འབད། དོ། ཡོདཔ། ཨིན། པས།

whereas the same input sentence with Tsheg 0F0C is incorrectly segmented as follows:

སངས་རྒྱས་དང་ཚེ་རིང་གཉིས་༌། བརྡ་དོན་དང་འཕྲུལ་རིག་ནང་༌།
ལཱ་འབད་དོ་ཡོདཔ་ཨིན་པས།

There are also cases like shortening of words, removing of inflectional words and abbreviating of words for the convenience of the writer. But this is not so reflected in the dictionaries, thus affecting the accuracy of the segmentation.

Following words has a special abbreviated way of writing a letter or sequence of letters at the end of a syllable as

རྡོ་རྗེ as རྡོྗེ

ཡེ་ཤེས as ཡེིས

etc..

6 Conclusion and Future works

This paper describes the initial effort in segmenting the Dzongkha scripts. In this preliminary analysis of Dzongkha word segmentation, the preprocessing and normalizations are not dealt with. Numberings, special symbols and characters are also not included. These issues will have to be studied in the future. A lot of discussions and works also have to be done to improve the performance of word segmentation. Although the study was a success, there are still some obvious limitations, such as its dependency on dictionaries/lexicon, and the current Dzongkha lexicon is not comprehensive. Also, there is absence of large corpus collection from various domains. Future work may include overall improvement of the method for better efficiency, effectiveness and functionality, by exploring different algorithms. Furthermore, the inclusion of POS Tag sets applied on n-gram techniques which is proven to be helpful in handling the unknown word problems might enhance the performance and accuracy. Increasing corpus size might also help to improve the results.

Acknowledgment

This research work was carried out as a part of PAN Localization Project (http://www.PANL10n.net) with the aid of a grant from the International Development Research Centre (IDRC), Ottawa, Canada, administered through the Center of Research in Urdu Language Processing (CRULP), National University of Computer and Emerging Sciences (NUCES), Pakistan. The research team would also like to express the gratitude to all the PAN Localization Project members of Bhutanese team based at Department of Information Technology and Telecom, for their efforts in collecting, preparing and providing with the lexicon, corpus, useful training and testing materials and finally for the their valuable support and contribution that made this research successful.

References

Chen, Stanley F., Joshua Goodman, 1998. *An Empirical Study of Smoothing Techniques for Language Modeling*, Computer Science Group, Harvard University, Cambridge, Massachusetts

Chiang, T-Hui., J-Shin Chang,, M-Yu Lin, K-Yih Su, 2007. *Statistical models for word segmentation and unknown word resolution.* Department of Electrical Engineering , National Tsing Hua University, Hsinchu, Taiwan.

Chungku., Jurmey Rabgay, Gertrud Faaβ, 2010. *NLP Resources for Dzongkha.* Department of Information Technology & Telecom, Ministry of Information & Communications, Thimphu, Bhutan.

Durrani, Nadir and Sarmad Hussain, 2010. *Urdu Word Segmentation.* Human Language Technologies: 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, June 2010.

Emerson, Thomas. 2000. *Segmenting Chinese in Unicode.* 16th International Unicode conference, Amsterdam, The Netherlands, March 2000

Haizhou, Li and Yuan Baosheng, 1998. *Chinese Word Segmentation.* Language, Information and Computation (PACLIC12), 1998.

Haruechaiyasak, C., S Kongyoung, M.N. Dailey, 2008. *A Comparative Study on Thai Word*

Segmentation Approaches. In Proceedings of ECTI-CON, 2008.

Huang, X., A. Acero, H.-W. Hon, 2001. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development (pp. 539 – 578).* Prentice-Hall Inc., New Jersey 07458.

Huor, C.S., T. Rithy, R.P. Hemy, V. Navy, C. Chanthirith, C. Tola, 2007. *Word Bigram Vs Orthographic Syllable Bigram in Khmer Word Segmentation.* PAN Localization Working Papers 2004 - 2007. PAN Localization Project, National University of Computer and Emerging Sciences, Lahore, Pakistan.

Jurafsky, D., A. Acero, H.-W. Hon, 1999. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (pp. 189 – 230).* Prentice-Hall Inc., New Jersey 07458.

Nugues, P.M. 2006. *An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with Special Consideration of English, French, and German (Cognitive Technologies) (pp. 87 – 104).* Springer-Verlag Berlin Heidelberg

Pong, L.W. and Robert. 1994. *Chinese word segmentation based on maximal matching and bigram techniques.* Retrieved from The Association for Computational Linguistics and Chinese Language Processing. On-line: http://www.aclclp.org.tw/rocling/1994/P04.pdf

Sunthi, Thepchai. 2007. *Word Segmentation and POS tagging.* ADD-2 Workshop, SIIT, NECTEC, Thailand.

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