A Computational Multilingual Text Constituent Splitter and Phrasing: A Case of Pashto Language

Zaheer Ahmad, Mohammad Abid Khan, Jehan Zeb Khan Orakzai, Rahman Ali, Ibrar Ahmad
Department of Computer Science, University of Peshawar, Khyber Pakhtunkhwa, Pakistan
ahmad.zaheer@yahoo.com, abid_khan1961@yahoo.com, janzeb@yahoo.com
rahmanali.scholar@gmail.com, toibrar@yahoo.com

Abstract

We propose a simple rules embedded matrix based method to split input sentences into their constituents and phrases. Splitting a sentence into phrases is a preprocess of machine translation for overcoming the problem of handling long sentences and improving quality of automatic translation. An effort is made to remove or at least minimize the problem of recursion that is faced during the process of phrase splitting thereby saving a lot of time. The system is dynamic in design and theoretically would work for any language that has some type of word order. However we have tested the system on Pashto language and this paper would describe the system in the perspective of Pashto language. The system can achieve more than 90% results keeping in view the Phrase Rules are carefully captured in a table.

1. Introduction

Sentence splitting and getting constituents is a specialized area of chunking. Sentences splitting into constituents are an important preprocess in a wide variety of NLP disciplines, particularly in machine translation and sentence generation. It is not only helpful to overcome the problems of complexity faced during the analysis of long sentences and improve the quality of translation. It can also be used in the translation process directly for complete phrases. This work is based on a language neutral approach to split sentences and get their constituents. However this research paper would mainly focus on Pashto language. This section is dedicated to introduce and to elaborate sentence analysis and recursion. In section-2, related work has been discussed. Section-3 is describing some challenges associated with Pashto language with the particular focus on Unicode for Arabic script. A detail discussion has been provided in section-4 about the Pashto language syntax rules. Section-5 is about the proposed approach to split sentences into their constituents. In the last, summary of the work has been shared.

1.1. Constituent Related Sentence Analysis

A group of words normally functions as a syntactic unit/ constituent/phrases in a sentence [1], [2], [3], [4], [5]. These groupings often give meaning to a sentence and help to identify important information about the structure of constituent boundary, linear order and syntactic categories [1], [2]. These constituents and Phrases can be substituted and replaced, moved in a sentence, deleted, merged and built up by a series of merger operations to form a larger constituent [1], [2], [3]. The order of the constituent in a sentence is as important in the syntactic study of a sentence as the word order [4]. It helps to understand a sentence after removing complexities, helps in translating a sentence and representing the structure of constituent. To model and represent constituent structure, Context Free Grammar (CFG) or phrase structure grammar has been successfully used for languages such as English [6]. However, there are many disadvantages in using CFG for natural languages such as ambiguity, left-recursion and repeated parsing of sub-trees. If a sentence is structurally ambiguous, then the grammar assigns to it more than one parse tree. It will be difficult to use CFG in languages that do not follow strict word order [6].

1.2. Recursion in Syntactic Structure of Sentence

Unlike a chunk, a constituent of a particular category can be embedded inside another constituent of the same category, which, in turn, can be embedded inside another such constituent. This property or set of

properties of a sentence are called recursion [7], [8]. Rules that govern these properties are called recursive rules [7]. In recursion the categories like NP or VP repeat itself on the right side of the arrow when written in Context Free Form. For Pashto language some examples are given in section 4.4. Usually recursion is graphically depicted in a tree form as shown in figure-I below. As in the figure-I, given below, when one NP or PP has another NP inside it then the first or head NP or PP is called a possessor phrase [2].

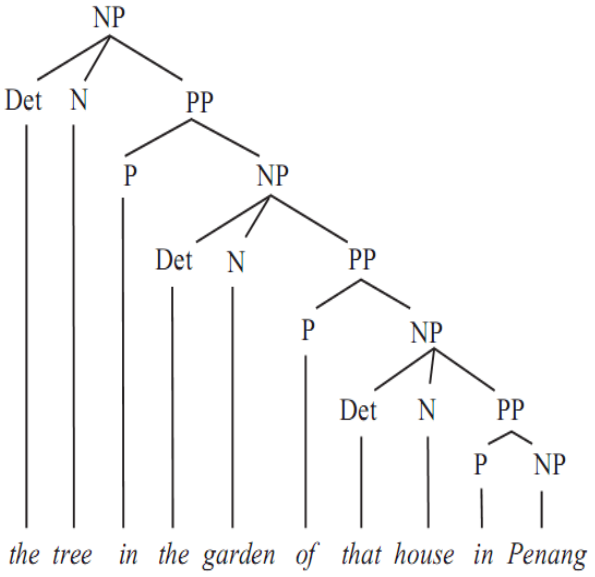

Figure I: Recursion

Pashto being a mixed word order language would not be suitable for processing by CFG because then one has to encounter and code a significant number of grammatical rules. However some special treatment of the same may produce good results. In this paper, CFG is used in the form of matrix to overcome the problems mentioned above.

2. Related Work

Sentence splitting or phrasing is usually carried out as a preprocess of machine translation to remove ambiguity from sentences and sometimes to generate sentence [9], [10], [11]. Different researchers have proposed different techniques. They are using parsing, collecting parts of trees as sentence, POS tagging and head verb or nouns to make phrases. In [10] Constraint Synchronous Grammar (CSG) has been used for this purpose. To identify NPs, Miorelli [12] proposed an ED-CER System for extracting noun phrases from Portuguese sentences based on a parser and a set of Noun grammar rules defined by Perini [13]. However, this approach needs sentences to be manually tagged. Costa [14] used LXGram to describe the grammatical properties and the meaning of Portuguese NPs. In [11], a tree based approach is used to generate phrases by identifying lexical properties of the head verb and the definiteness of arguments and their length. On the other hand, some, authors have used statistical approaches as in [15] statistical recognition of noun phrases with a chunk tagger is used, and it is presented that a part-of-speech tagger can be used for phrasing.

3. Challenges in Pashto Language

Pashto being a low resourced language [16], [17] presents many challenges. Some of them are presented here. First of all, a problem with its Unicode block would be discussed.

3.1. Unicode Problems of Pashto Language

While developing the software for the constituent splitting, it was needed to match Pashto text with words in a lexicon. During the process it was revealed that a word could not be matched in the lexicon despite it is there. On further investigation, the authors came to know that the characters in the target word, though looking similar and same are using different Unicode than the word from the corpus. It was found that different text editors were using different Unicode from the Arabic block of characters, as there are characters that look similar but have different Unicode.

Pashto script comes under the Arabic block of Unicode like many other languages e.g. Urdu, Persian, Panjabi, Sindhi, Balochi and Kashmiri. In Arabic Unicode, a number of duplications are introduced as stated in [18] about Urdu Unicode that as Unicode standard has to cater to multiple existing systems and multiple languages within a script, redundancies are introduced in it. It is advised to re-standardize and short list the characters nationally before usage. However it seems that during designing and encoding, unification of characters have not been duly cared. As it is mentioned in [19] that some extended Arabic characters are typographical variants of characters already fully covered by the corresponding basic Arabic characters. In [19], it is further stated that in some cases it looks that Unicode - knowingly - confuses regional calligraphic or typographic variants for encodable characters. Some justifications are given for the existence of a number of "goal yeh" there in the Unicode but people knowing the languages like Urdu,

Arabic and Pashto feel confused and some changes are needed to bring unification in the Arabic Block of Unicode, in order to make it easy for developers as well as for common users. As the Unicode standard [20] clearly states that all similar characters would be unified across languages.

Pashto is natively spoken in a number of countries such as Pakistan, Afghanistan and other parts of the world like the Gulf states, Europe, UK, India and America [21], [22], [23], [24]. Therefore, standardization for each country might differ from the other, then how data transformation would take place. Furthermore, what would be the solution for usage of these scripts on the net and which country's standard would be followed? Therefore, nation wise standardization is not feasible. It is also not a good idea to further devise standards within a standard. It would lead to the situation that existed before Unicode and too many standards for one language script would lead to no standardization at all. In works like [25], the author presented some solutions that need further development. However in the present scenario, despite the Unicode, researchers feel as if there is no standardization for the Arabic based scripts such as Pashto and Urdu. The authors believe that significant achievements can be made through hinting codes of the fonts for each language without using duplication of similar characters.

3.2. Variance in Spellings of Pashto Language

There are a number of dialects of the Pashto language. An example is the word Pashto itself as Pashto can be written and spoken as Pakhto, Pushto, Pukhto, Pashtu, Pakhtu, Pushtu, Pukhtu, Pukkhto, Pukshto, Paktu, Pooshtoo, Passtoo,Pakhtoo, Pakkhtoo and Pasto [16]. The most common characters that are used interchangeably are Kh as sh, o as u and o as oo. These characters offer many challenges while comparing strings therefore one should take care of these issues beforehand. In this paper the authors worked hard to overcome this problem by coding separate module for this.

3.3. Morphology of Pashto

Pashto has many inflectional forms in its major categories such as nouns and adjectives are differentiated for case, number, and gender [17]. However till date no complete work has been presented to capture all morphological variations. In addition, nouns are not necessarily the same class or gender for different speakers, and occasionally there is even variability within a speaker.

4. Pashto language, Syntax and Phrase Rules

Pashto is the language of over 20 million people. Some claim it to be 40-60 million. It is mainly spoken in Pakistan and Afghanistan and has more speakers in Pakistan than Afghanistan [21], [16], [24]. It is also spoken in other parts of the world like in the Gulf, Europe, UK, India and America [23], [22]. Despite that Pashto is a low resourced language [16], [17] and offers many challenges in terms of its complex syntax and phrasal rules.

Pashto is fairly rigidly head-final in NP and VP lexical categories, while several functional categories are head-initial [21]. The basic word order is SOV with some degree of word order freedom and split-ergative language [21]. These and other properties require a considerable amount of effort to capture the phrase structure rules for the language. Numerals and adjectives precede any nouns they modify, suggesting that the lexical category NP is head-final [21]. It is specifically only the lexical projections (VP, NP) that are head-final. With regard to the PP projection, the language appears to exhibit mixed headedness [21]. Some phrasal rules extracted from [26], [27], [28] are given in the subsections below one by one.

4.1. Noun Phrase

- A Noun Phrase consists of a noun or a pronoun together with modifiers that may be an adjective.
- An adjective usually precede a noun but it can appear after noun depending on the context.
- A noun can precede a preposition [phrase].
- The order of the modifiers may be like Preposition + demonstrative(that)+ quantifier + indefinite article(some, a)+descriptive adjective (big, pretty) + noun.
- Adverb that modifies adjectives (very) occurs immediately before adjectives they modify but the order can be altered if the speaker wishes to focus/stress one or the other of the modifiers.

4.2. Verb Phrase

- A verb phrase includes everything except the subject

- Verb is usually the last word in a sentence
- Usual order is a time phrase + Complement/ object+ place phrase + other modifiers + verb
- If object of a preposition is a weak pronoun, the prepositional phrase is almost always positioned just before the verb.
- In negative verb phrase, the negative article " na " occurs before the verb in the imperfective tenses
- In perfective tenses the negative article "na" occurs with simple verbs between the perfective marker and the verb stem.

4.3. Adj Phrase, Adv Phrase and PP Phrase

- Prep comes after and/or before noun or both before and after.
- Adj always comes before noun. Adj can be used as noun.

4.4. Context Free Grammar of Pashto

Based on the work in [29] and the above rules the Context Free Grammar for Pashto is given as below:

- S --> NP+VP|VP|NP+CONJ+VP
- NP --> N|PN |ADJ+N|CN|NP+PP| NP+NP | PP+CONJ+PP | PN | ADJP+ NP | N | NP+ VP | PRON | PP | Det+Adj+NNP+V | Det+Adj+N+VP | Det+Adj+N+Adv+VP
- VP --> V |VP+VP |NP+VP PP+VP|AUX|PPP+VP|PREP+VP|PP+VP|ADJ+VP|ADV+ADJ+V|ADVP+VP |V+NP| NP+V| Adv]+ PP +V
- PP --> PREP+NP |PREP+NP |PP+PP |PREP+NP POSP |PREP+NP |PREP+VP |PREP+N |NP+PP
- ADJP --> ADJ |ADJ+N |ADJ+ADJ

5. The Proposed Methodology

The proposed algorithm is robust and dynamic. It can split a sentence into constituents by taking the dictionary that has grammatical categories in separate column, a matrix having syntactic / word order rules of a language and the corpus from which each sentence has to be splitted into constituents. The table having rules is shown in section 5.3. The complete architecture of the system is given in figure-II, below.

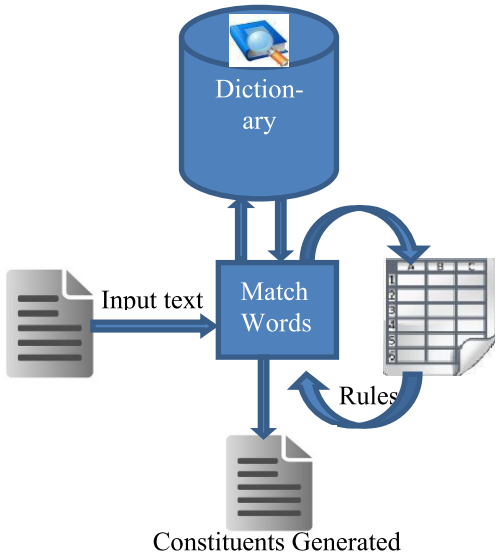

Figure-II. System Diagram

All the rules were developed in a table in MS Excel, lexicon was saved in a database format in MS Access, where as the corpus was in plain text format. C# (Visual Studio) was used to develop and test the system.

To understand the working of the proposed system, a step wise flow of the system is given in the following subsections.

5.1. Detecting words and the end of sentences

A sentence is read character by character from a text file, and wherever a white space is found a word is marked there. A dot (.) or (-) or (?) is marked as the end of the sentence. The system is developed for singe words only and multiwords are not considered because of the unavailability of a lexicon for the same.

5.2. POS Tagging of the Words

A lexicon of Pashto language having more than 14000 words is used by the software. The dictionary is made of a table, having Pashto word, POS and meaning of the word in English in separate columns. On reading the text from the text file, each word is matched in the dictionary with its list of words. Matching a Pashto word with the dictionary suffers badly from the Unicode problems as discussed earlier in this work. Some extra lines of code have been written to solve this problem and the problem of the different dialects of the Pashto language. To overcome the Unicode problem, all similar characters with different Unicode are stored

in a two dimensional matrix to check alternate Unicode for a character if a word is not found in the lexicon attached with the software. During the process when a word is found in the lexicon, its lexical category is read from the relevant POS column. Otherwise the module for Unicode and dialect is called to execute and find the relevant word in the lexicon. If still no word is matched in the lexicon, the word is marked as 'Unknown'. Below figure-III, shows a screen shot of the POS tagged list of words taken from the software.

	PashtoWord	WordCat
▶	افغان	Noun
	چارواکي	Unknown
	وايي	Unknown
	چې	Adverb
	دغه	proNoun
	سند	Noun
	به	particles
	تر	particles
	پنځو	Unknown
	کلونو	Unknown
	پورې	Adverb
	اعتبار	Noun

Figure III: System Generated POS Sample

5.3. Rules embedded in a table

Rules described in section-4 of this paper are embedded in tabular format in order to get constituent of a sentence. A partial list of these rules IS given in Table-I.

Rules from table-I are read from top to bottom, and left to right, starting from titled column P1 of the table. Column under P1 are read, lexical categories for its relevent rows in the POS titled column are counted and read. The relevent entries in the POS columns are matched against the tagged text in figure-III. If the POS of the read text are matched with the lexical elements of the POS column for the P1 column, it means the combination of words against the tagged POS is a valid phrase or constituent. The process is rpeated for each column of the Table-1 untill S titled column is reached. This matching process is linear in order however the constituents occurs recursively in natural text. Therefore the text is read linearly however each found constituents is marked with its position in the sentence. Whenever another constitutent is found in the same position, these constitutnets are placed in order , as they were before. This way the problem of recursion is sloved and the shorcoming of CFG is overcomed.

Table 1: Syntax and Phrase Rules

S#	PoS	P1	P2	P4	P5	S
1	Noun	NPP	DAN	X2	Noun Phrase	S
2	*PP*					
3	Noun	NAP				
4	Adv		NA			
5	proN					
6	Noun		proN	X3		
7	Part	PA	PAVP		Verb Phrase	
8	Part					
9	Unkn	Unkn				
10	Unkn		ADP			
11	Adv	Adv				

5.4. How Recursion is tackled

As discussed earlier in section 1.3, recursion is the property of a constituent to contain another constituten. A sentence may have many levels of constituent within constituent. Opening up this layer by layer is not an easy process. Parsing and syntactic tree are mostly used to catch these layers. However, here in this work, the phenomena of recursion is tackled as an iterative process to simplify the complexities of recursion. As given in the algorithm below, during the process of matching of rules against the tagged words/sentence, the rules check only for constituents without looking for constituents within constituents. However a log is maintained of the location of each constituent with in the sentence or possessor constituent by keeping a count for each word. This log is used in the end of sentence completion to put all the constituents in an order.

5.5. Algorithm

The proposed algorithm to read the un-tagged corpus, tag the words of the corpus using a lexicon and make phrases based on rules embedded in a table is given below.

1. **Read text** from un-tagged Corpus word by word
2. **Search Dictionary** for each word read in step-1
3. **If Match Found** read its relevant grammatical category from the dictionary attached

4. **ElseIf no match found**, repeat step-3 with different Unicode for the same word (as some characters repeat in the Arabic Block with different Unicode)
5. **ElseIf No Match Found** repeat step-3 with alternate characters(form) for the same word(as in Pashto some words have the same meaning with different form in terms of spellings e.g. Pashto is written both as Pashto and Pakhto)
6. **Match Found**, then TAG each matched word with its grammatical category
7. **No Matched Found**, TAG the word as 'Unknown'
8. **Read Rules** attached, from the matrix
9. **loop step 9-13** to fire rules from top to bottom and left to right
10. **Apply Rules** on the list of words tagged with grammatical category
11. **If Passed The Rule** by the set of words then,
12. **Look for old location,** write the new phrase and old phrase together and mark the position of new phrase
13. **Increment** to change set of words or rules
14. End

The main advantage of the proposed algorithm is its speed and ease of use. The table has been used to work like "if-then-else" clauses or rules. Coding and firing of rules is not only a complex and tedious job but also suffers from the recursion when dealing with analysis of sentence structure. Whereas rules embedded in a table like in the proposed algorithm makes the whole process recursion free, faster, and easy to build, change and improve rules.

6. Summary

The system is proposed to split input sentences into their constituents and phrases. A simple knowledge based system having all rules in a table is presented in this paper. The splitting process is quite encouraging with more than 90% results for any language. The algorithm used is mainly tested on Pashto language. The algorithm is designed to minimize the pitfall of CFG and overcome the complexity arises because of lengthy sentences.

References

[1] Paul R. Kroeger, *Analyzing Syntax: A Lexical-Functional Approach,* Cambridge University Press, New Yoark, 2004.

[2] Paul R. Kroeger, *Analyzing Grammar: An Introduction*, Cambridge University Press, New York, 2005.

[3] Andrew Radford*, Analyzing English Sentences a minimal approach*, Cambridge University Press, New York, 2009.

[4] Maggie Tallerman, Hodder Education*, Understanding Syntax (third Ed)*, UK, 2011.

[5] Bas Aarts*, Syntactic Gradient the nature of grammatical indeterminacy,* Oxford University Press, New Yoark, 2007.

[6] Model Selvam M, Natarajan. A M, and Thangarajan R, "Structural Parsing of Natural Language Text in Tamil Using Phrase Structure Hybrid Language", International Journal of Computer and Information Engineering 2:4 2008.

[7] Steven Bird, Ewan Klein, and Edward Loper , *Natural Language Processing with Python,* O'Reilly Media, Inc, 2009.

[8] Santorini, Beatrice, and Anthony Kroch, "The syntax of natural language: An online introduction using the Trees program", 2007. Available: http://www.ling.upenn.edu /~beatrice/syntax-textbook.

[9] Takao Doi, Eiichiro Sumitta , "Input Sentence Splitting and Translating: ATR Spoken Language Translation Research Laboratories Hikaridai, Kansai Science City, Kyoto, Japan.

[10] Francisco Oliveira , "Systematic Noun Phrase Chunking by Parsing Constraint Synchronous Grammar in application to Portuguese Chinese Machine Translation" , in proc. ICITA, 2009.

[11] G. Kempen and K. Harbusch, *"Generating Natural Word Order in a Semi-free word order Langauge: Treebank-based linearization preferances for German",* in proc.. *of the Fifth CICLING, Seoul, Korea 2004.*

[12] S. T. Miorelli, "ED-CER: Extração do Sintagma Nominal. em Sentenças em Português", Ph. D. Thesis, Pontifícia Universidade Católica do Rio Grande do Sul.

[13] M. Perini, "A Gramática descritiva de Português", São Paulo: Editora Ática, 1995.

[14] F. N. Q. M. C. Costa, "Deep Linguistic Processing of Portuguese Noun Phrases", Master Thesis, Universidade de Lisboa, Portugal.

[15] Wojciech, S. and Thorsten, B. “Chunk Tagger - Statistical Recognition of Noun Phrases”, ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing, Saarbrucken, 1998.

[16] Craig Korpis, “Computing in Pashto An overview of a major language in Afghanistan and Pakistan”, in proc. *Multilingual Computing & Technology.*

[17] Andreas Kathol, Kristin Precoda, Dimitra Vergyri, Wen Wang, Susanne Riehemann , *“Speech Translation for Low-Resource Languages: The Case of Pashto”* in Proc. Interspeech, 2005.

[18] Sarmad Hussain, , Sana Gul, Afifah Waseem ,”Urdu Encoding and Collation Sequence for Localization” , Center for Research in Urdu Language Processing National University of Computer and Emerging Sciences.

[19] Tom Milo, *Some comments on the Arabic block in Unicode*, DecoType.

[20] Unicode Standard 6.1, Date, Retrieved 06/06/2012, Available:www.unicode.org/ versions/Unicode6.1.0/References.pdf

[21] Taylor Roberts, *Clitics and Agreement,* Phd thesis for Doctor of Philosophy in Linguistics at the Massachusetts Institute of Technology ,June 2000

[22] Hamza A Shierani, *Building bilingual Anglo-Pashto Proper noun Lexicon from the we*b, Thesis, Department of Computer Science, The University of Shefield

[23] Lewis, M. Paul, Pashto, Southern: A language of Pakistan**,** Date, Retrieved: 6/6/2012, Available: http://www.ethnologue.com/ show_language.asp?code=pbt

[24] Penzl, Herbert; Ismail Sloan, *A Grammar of Pashto a Descriptive Study of the Dialect of Kandahar, Afghanistan*. Ishi Press International. 2009.

[25] Jonathan Kew, *Notes on some Unicode Arabic characters: recommendations for usage* (Draft), Retrieved:6/6/2012, Available: http://scripts.sil.org/cms /scripts/render_download.php?format=file&media_id=arabicletterusagenotes&filename=ArabicLetterUsageNotes.pdf

[26] Tegey, Habibullah, Robson, Barbara, *A Reference Grammar of Pashto,* Washington, Center for Applied Linguistics, 1996.

[27] Zarghona Rekhten Zewar, Pakhtu Nahoa (Grammar), Da Sapye Da Pakhto Serono ao Prekhtia Zaye Markaz,Peshawar, 2003.

[28] Sidiq Ullah Reshteen, Da Pakhto Ishtiqaqona ao Tarkebona. Peshawar.

[29] Rahman Ali, Muhammad Amir, Mohammad AbidKhan “Developing Pashto Treebank”, in Proc: International Conference on Networks & Information Technology (ICCNIT), University of Peshawar, Pakistan, 2011.