Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (1) 43-48(2012)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

A Model for Sindhi Text Segmentation into Word Tokens

J. A. MAHAR, H. SHAIKH*, G. Q. MEMON

Faculty of Engineering, Science and Technology, Hamdard University, Karachi
E-mails: mahar.javed@gmail.com, hidayat.shaikh@salu.edu.pk, gqmemon@yahoo.com Ph: +92-334-2727937

Corresponding author Javed A. Mahar, E-mails: mahar.javed@gmail.com

Received 12rth August 2011and Revised 10ndJanuary 2012)

Abstract: The corpus is prerequisite to conduct the experiments of computational linguistic applications on any language. Generally, the corpora are downloaded from Internet in different formats. Usually, the downloaded corpora have some types of word ambiguities regarding computational processes; however, it is observed that in Sindhi language, two types of ambiguities are commonly found i.e. compound words typed without embedded space and typo errors. Without correct segmentation of text into word tokens, it is difficult to get better results of linguistic applications. Therefore, tokenization is the inevitable component of natural language and speech processing applications. This paper presents a new model that correctly segments the words of Sindhi language. The model consists of three layers; layer 1 is used to input the text and segment the words using white space, simple and compound words are segmented in layer 2 and complex word are segmented in layer 3. The tokenizer is tested on 2792 Sindhi words and it achieved the accuracy of 91.76%.

Keywords: Sindhi Language; Tokenization; Corpora; Layers; Word Ambiguities

1. INTRODUCTION

Segmentation of input sequence of orthographic symbols is called tokenization. For language modeling, the distribution of input text into tokens is compulsory. Concatenation of syllables is the base of Sindhi orthography; on the contrary, the white spaces between words, like in English, delimit them. Some sorts of words bring about ambiguity for the segmentation process because there is an embedded space within them. For instance, a word پس پرده (off the screen) is one of such compound words and a deliberate hard space is put between these two; (pasa-parda) so this affects the process of tokenization.

The corpus of the language is necessary for development of the linguistic applications for instance, parts of speech tagging (Sajjad, *at el.,* 2009), morphological analysis (Dror, *at el.,* 2004), text to speech (Mahar, *et al.*, 2010) and diacritics restoration (Mahar, *at el.,* 2011). Generally, the corpora are downloaded from web in DOC and PDF formats and converted into equivalent plain text. It is observed that the corpora contain usually two types of word ambiguities in Sindhi, i.e., compound words without embedded space and typo errors. Without correct segmentation of text into word tokens, it is difficult to get the better results from the above mentioned applications. Therefore, tokenization is the inevitable component of many natural language and speech processing applications.

Since last few years, several models have been proposed for tokenization of Arabic script and other scripts of Western and Asian languages. Various research papers have been published in which the tokenization models of Arabic script based written languages like for Urdu (Iiaz and Hussain, 2007, Durani and Hussain, 2010) and Arabic (Habash and Rambow, 2005, Attia, 2007). These proposed tokenization models can be used for Sindhi also but at an extent, each language owns different sets of rules. Therefore, a new Sindhi tokenization model is proposed in this paper. This proposed tokenization model has enough efficiency to deal with all the words ambiguities that create problems in the segmentation process of Sindhi language.

2. MATERIALS AND METHODS

Material:

The development of corpus is essential for computational exposition of a language. Sindhi language owns the capacity of adaptation of words of other languages; this is also an important feature of a language to meet the modern standards of communications. The corpus of the published books in Sindhi is collected from various genres, i.e., story,

* Department of Computer Science, Shah Abdul Latif University, Khairpur, Pakistan

linguistic, and history. Most of the corpus is manually developed and some data was also converted into plain text equivalent, from other formats, like PDF and HTML. The corpus contains approximately 108,556 words. The detailed information of corpus is given in (**Table 1**).

Table 1: Words Information of Developed Corpus

Book	Genre	Sentences	No. of Word
Sindhi Viya Karan	Linguistic	2,951	29,185
Dil Jee Duniya	Story	4,408	41,374
Sindh Jee Fattiha	History	3,765	37,997
Total		**11,124**	**108,556**

Method:

Basically, tokenization is used as a component of many natural language processing applications. Every language for its computational development needs its own tokenization model. Hence, the method varies from one to other language. In our developed Sindhi tokenization model, the method is divided into three tokenizing layers. This division has been based on the types of words in Sindhi language; hence, each layer deals with its specified job depending on the type of word.

Sindhi Tokenization

The process of segmenting input sequence of orthographic symbols into tokens is called tokenization (Attia, 2007). After that these tokens are supplied to natural language processing applications for further computational processing. The word boundaries like white space, punctuation marks, digits and special symbols are used for tokenization.

Sindhi language has many ambiguities for tokenization of words. There are different types of words that bring ambiguity to the tokenization process. The compound word that has to be taken as a single word but the embedded spaces needed in between are the case for segmentation process of tokenization. The letters are of two types in Sindhi; connector and non-connector, they also create complication and need more attention. The ambiguities observed in general compositions ought to be analyzed prior to develop a tokenizer, hence, the ambiguities occurring frequently in Sindhi text, are described in the next section.

Tokenization Ambiguity

Concatenation of syllables is the base of Sindhi orthography; on the contrary, the white spaces between words like English delimit them, for instance, احمد منهنجو پٽ آهي. (Ahmed is my son), there is no any ambiguity because each word of this sentence is delimited by white space. Different morphological issues exist in Sindhi where this approach has failed due to some sort of words that bring about ambiguity for the segmentation process because there is an embedded space within them, particularly in compound words. Consider the word سنڌوندي (Indus River), a soft space is seen in between the word سنڌو (Indus) and the word ندي (River), both are separate words in the lexicon.

In Sindhi alphabet, there are two types of characters: connector and non-connector. Connector characters automatically joint with next character, like ب، ل whereas non-connector characters are not in joint with succeeding character, like د، ڏ. The character و is also a non-connector character so that a soft space is seen in word سنڌوندي (Indus River). In Sindhi text editor, system does not store this soft space in memory. Therefore, this type of space does not create problem in tokenization. On the other hand, if the ending characters are connectors then writer must insert a hard space between two parts of word to avoid them from joining and consequently to maintain the separate ligature identity. Consider the word اسلام عليڪم here the space between اسلام and عليڪم explicitly inserted otherwise this word is visualized like اسلامعليڪم and such formation is incorrect in the language. This explicit space affects tokenization process because system creates two tokens of a single word. This problem also creates ambiguity in natural language processing applications like in parts of speech tagging. For example, word صاحب قدرت is an adjective. This is a compound word, containing two words صاحب /Lord/ (noun) and قدرت /Nature/ (noun). If this word is written without explicit space then it looks like صاحبقدرت. As we already have mentioned that connector-characters automatically join the next characters if space is not put in between. Therefore, in case of hard space, system would erroneously tag such single word with two tokens.

The two special words ۾ (in) and ۽ (and) also create ambiguity because both words are normally typed without preceding or following space with other words, for instance, (mounmain), however (moun) and (main) are two different words but tokenized as single word. Similarly, ميز۽ڪرسي (table and chair) are tokenized as single word but actually three words are written without space (Rahman, 2010). The words with non-connective endings like first word of these two; ڪ يرڀ ي (drink milk) and starting characters of suffix morphemes like second word of these two; سنڌاندر (in Sindh) also create ambiguity for tokenization.

Proposed Model for Sindhi Tokenization

Various solutions have been proposed to solve the problem of words segmentation of a text into

tokens for natural language and speech processing applications in Arabic script based languages. Sindhi, a superset of Arabic alphabet, is also one of the Arabic script based languages. The models proposed for such languages can also be experimented on Sindhi but, at a considerable extent, every language differs from other in terms of grammatical, orthographic and linguistic rules. Therefore, a Sindhi tokenization model, depicted in (**Fig.1**), is proposed in this paper. The proposed model overcomes all the above discussed ambiguities in Sindhi text. It consists on three layers which have their special efficiency to deal with the different types of word ambiguities found in Sindhi language. Each layer performs its function individually for every type of words hence it depends on the type of input word that how much process is necessarily required for its segmentation. Each word needs to go for further than layer 1, and then system automatically shifts to the next layer and so on up to 3rd layer. Therefore, a stratified combination of three layers makes a state-of-art of tokenization.

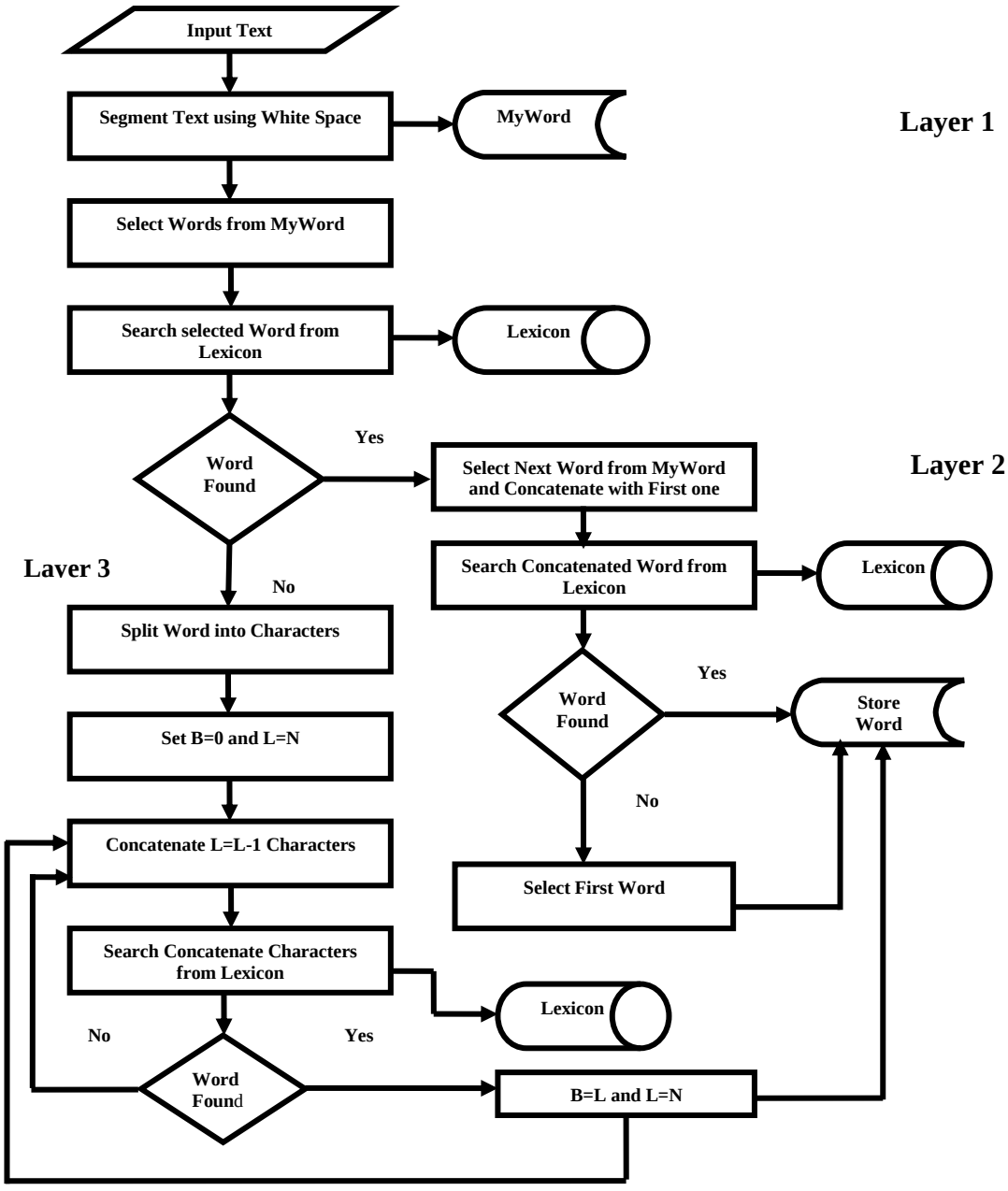

Fig.1. Proposed Tokenization Model

3. IMPLEMENTATION AND RESULTS

Three types of words can be seen in routine applications which are simple, compound and typo-errors. The developed corpus is used for the experiments through our proposed model of Sindhi tokenization. The model takes corpus as input and converts it into word tokens by using three layers. Simple and compound words can be tokenized by layer 2 where as other standard words having typing mistakes or non-standard words are tokenized by layer 3. All word tokens are stored into separate databases for further processing. Each layer with its implementation process is described below.

Layer 1: This layer is particularly designed to input the next, segment it and search the matching word from developed lexicon; this lexicon contains simple and compound words only. In detail, system takes input text from user or file and segments it into words using white space and stores in the array MyWord. Then one by one system selects words from this array and searches each one from the lexicon. In case, the word is present, the system proceeds to layer 2, otherwise refers to layer 3.

Layer 2: At this layer, only simple and compound words are segmented into tokens, for this, system selects next word from the array MyWord and then concatenates it with the previous word. Then system searches the concatenated word from lexicon to ensure whether it is a compound word or not, if system has found the word then stores it into database named Store Word and if word is not found from lexicon then system selects only first word and considers it as simple word, finally stores into array Store Word.

Layer 3: The words having typo-errors are dealt with at this layer so that reaching the 3rd layer; system splits word into characters and stores them into a temporary generated array. The variable B is used for beginning location of array and variable L is used for the ending location of the array therefore, initially we set B=0 and L=N. Next, system concatenates L=L-1 characters and then searches these concatenated characters from lexicon. In case, the word is present, the system sets B=L and L=N and stores word into database Store Word, otherwise again concatenates L=L-1 characters and continues till word is found.

Results

The cumulative accuracy of 91.76% is achieved on the developed corpus of 108,556 words. For testing, the text of 2792 words was randomly taken from the corpus; the words in the selected text had such types of ambiguities which bring the problem to the process of tokenization. The calculated accuracy of each type of words using proposed tokenization model is shown in (**Table 2**). The graphical representation of calculated accuracy is depicted in (**Fig.2**).

Table 2: Calculated Accuracy using Proposed Tokenization Model

Types of Words	Accuracy %
Simple Word	100
Compound Words	91.4
Words Having Typos	83.9

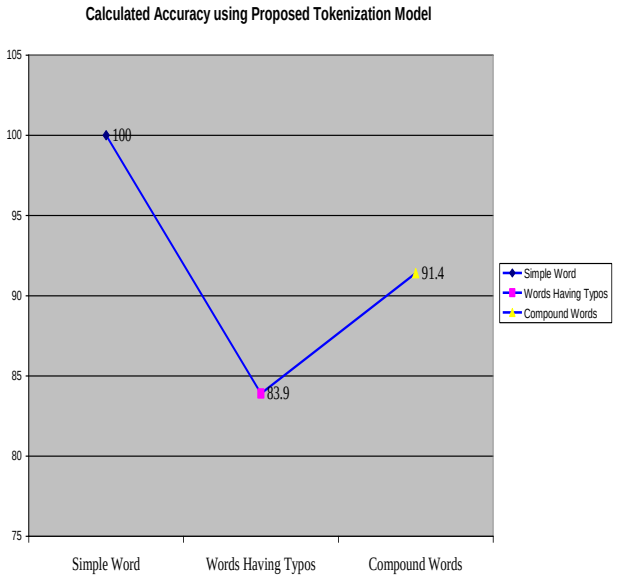

Fig.2.Graphical Representation of Calculated Accuracy Using Proposed Tokenization Model

The results shown above define that our developed tokenizer segments the simple words exactly. However, the performance with other types of words is lower than that of simple words because there are certain reasons and problems in Sindhi writing system that still persist in computational processes. The 91.4% accuracy with compound words is achieved due to the connecting letters occurring at the end of the first primary word that connect with the next first letter of the second primary word because no space is used in between two parts of a compound word. Hence, the tokenizer can not segment accurately sometimes in such situations. The accuracy of 83.9% with the words having typos denotes the capability of the developed tokenizer for it basically works at letter level. Actually, the tokenizer calculates the letters in descendant order and when the calculated letters form a word thus it segments them as a word. However, this method is better for Sindhi language but still needs improvement for more accurate performance.

4. CONCLUSION

The tokenization model is proposed for Sindhi language. The received results prove the success of this model for the sound tokenization of Sindhi text. This model stands as a necessary and inevitable prerequisite for many natural language and speech processing applications for Sindhi, for instance, diacritics restoration, text to speech synthesis and text recognition, it plays a key role before proceeding for further process. The model consists on three layers; layer 1 is used to input the text and segment the words using white space, layer 2 segments simple and compound words and complex words are segmented in layer 3. The tokenizer is tested on 2792 Sindhi words and it achieved the accuracy of 91.76%. This model can also be used on other Arabic script based languages like Urdu, Persian and Arabic because these languages have similarities to the orthographic system of Sindhi.

REFERENCES:

Attia, M. A., (2007) "Arabic Tokenization System", In the Proceedings of the Workshop on Important Unresolved Matters, Prague, Czech Republic, 65-72.

Dror, J., D. Shaharabani, R. Talmon, and S. Wintner, (2004) "Morphological Analysis of the Qur'an", Literary and Linguistic Computing, Vol. (**19**): Number 4, 431-452.

Durani, N., S. Hussain, (2010) "Urdu Word Segmentation, Human Language Technologies", The Annual Conference of the North American Chapter of the ACL, Los Angeles, California, 528-536.

Habash, N., and O. Rambow, (2005) "Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop", In Proceedings of the 43rd Annual Meeting of ACL, Ann Arbor, 573-580

Ijaz, M. and S. Hussain, (2007) "Corpus Based Urdu Lexicon Development", In the Proceedings of Conference on Language Technology, University of Peshawar, Peshawar, Pakistan.

Mahar, J. A., G. Q. Memon, and H. A. Shah, (2010) "WordNetBased Sindhi Text to Speech Synthesis System", Proceedings of the 2nd International Conference on Computer Research and Developmented Kuala Lumpur, Malaysia, 20-24.

Mahar, J. A., and G. Q. Memon, (2011) "Automatic Diacritics Restoration for Sindhi", Sindh University Research Journal (Science Series), Vol. (**43**): Number 1, 43-50.

Rahman, M.U., (2010) "Towards Sindhi Corpus Construction", Conference on Language and Technology, Lahore, Pakistan.

Sajjad, H., and H. Schmid, (2009) "Tagging Urdu Text with Parts of Speech: A Tagger Comparison", Proceedings of the 12th Conference of the European Chapter of the ACL, 692-700.