2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)

An Efficient Stop Word Elimination Algorithm for Urdu Language

Kamran Shaukat Dar*[1], Ahmad Bin Shafat[1], Muhammad Umair Hassan[2]
Department of Information Technology
University of the Punjab, Jhelum Campus
Jhelum, Pakistan
[kamran*[1], bcs.f12.04[1], bcs.f13.23[2]@pujc.edu.pk]

***Abstract*— Stop words occur multiple times in a document and the occurrence of stop words have least semantic value in the document sentences. These words cover a noteworthy bundle of archives that have no semantic significance. So, the stop words ought to be removed for better language description. In this paper, we have proposed a proficient algorithm which will eliminate the Urdu document stop words. Many considerable efforts have been performed in the areas like natural language processing (NLP), stemming for Urdu language and sentence limit disambiguation. However, there is no such work available for Urdu language that can remove the stop words from an Urdu document. That is the motivation behind this work that we proposed stop words elimination algorithm from Urdu language documents. This is being carried out for the first time in Urdu language by our proposed algorithm.**

Keywords—stop word; Urdu; natural language processing; stemming

I. INTRODUCTION

The words that occur commonly in a document have least significance in the context of semantic values are known as stop words [1]. These words are only used for grammatical restricted orders. Some of these words in Urdu language are: لی رہی،کہتر،تو،اور پر مرگ گی کی کا کی تھا تھی تھے، جو گے ے ل،، ے گی، چکا، چکی، چکے.

A. Urdu language details:

Urdu is described as an Indo-Aryan language and it is also categorized as a branch of Indo-European language. The syntax of Urdu is written in left-to-right order. There are about 300 million speakers of Urdu in all over the world and use Urdu as their native or secondary language [2,3,4,7]. "Urdu اردو " word is taken from a Turkish word "ordu".

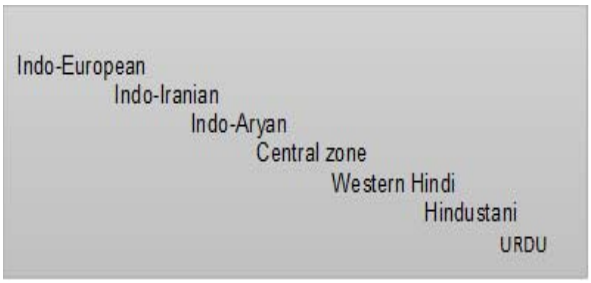

Fig. 1. Language Family Tree for Urdu [11]

The word "ordu" have a meaning which is "Tent" or "Army" or "horde". Urdu is also known as the "Lashkari Zuban (لشکری زبان)" i.e. "The Language of the Army" [2] due to the reasons provided above. The national language of Pakistan is Urdu. It is also a fact that in India there are twenty-three officially spoken languages and Urdu is one of them. Urdu is composed in Arabic and Perso-Arabic Script [2, 5]. Urdu language has much similarity with Hindi language. However, a big portion of this language consists of Arabic, English, Punjabi, Persian, Sanskrit, Turkish, and other languages [2, 9, 12]. Urdu language is inflectional morphological and logically rich text language [6]. Urdu is a source of communication in different countries like Afghanistan, Bangladesh, Bahrain, Botswana, Fiji, Mauritius, Germany, Guyana, India, Malawi, Nepal, Norway, Oman, Thailand, Qatar, South Africa, Saudi Arabia, UAE, United Kingdom, and Zambia [11]. Urdu is a sensitive language in the context of syntax. This is because in an Urdu text document the start, middle, and end of document differs with respect to its position and shape [2]. The letter گ is considered as letter for the sake of our example. This can be viewed in the table that in the words like جنگ, جنگل, گیا the letter گ has changed its shape with respect to its position.

Table I: Context-Sensitivity of Urdu language

گیا	جنگل	جنگ
Start	**Middle**	**End**

The paper is organized as follows: the text mining techniques have been elaborated first that are stemming, tokenization and stop sword elimination etc. After this an overview regarding this domain is presented and then different methodologies are explained in the context of stop word elimination algorithm for different languages. In the last section, we will propose an algorithmic methodology to eliminate Urdu document's stop words.

II. RELATED WORK

The process in which we get the useful information from text document data and process the information is known as text mining [13]. There are various techniques for text mining in many other languages.

978-1-5386-0449-6/17/$31.00 ©2017

A. Stemming

The process in whcih we lessen the derivationally related words into a typical base form and removal of inflectional words from document is called stemming [1].

B. Tokenization

This is linguistic pre-processing methodology, in which we deal with a way toward building the equality token classes and chopping down streams of characters into tokens [1].

A statistical stemmer was proposed by Larkey *et al* for co-occurrence based Arabic retrieval. The morphological analyzer was used to compare the statistical and light stemmers. Their stemmers worked efficiently for cross-language retrieval [8]. A Punjabi stemmer was proposed by Rajeev Puri *et al* for stemming of all Punjabi words. The revised suffix removal approach and extended stripping rules were used for their stemmer [10]. Assas-Band was proposed by Qurat-ul-Ain *et al*, which is an Urdu stemmer that is used to remove prefix and postfix. The form of stem is made when letters added by stemmer [6]. A template based stemmer was proposed by Sajjad Khan *et al* to perform a rich language morphological stemming. All kinds of affixes like prefix, infix, and postfix were removed by their stemmer. Also, the stemmer proposd by them gave the 89.05% precision, 92.49% FI-Measure, and 96.08% recall [2]. The problems occurred during Urdu text tokenization and sentence boundary disambiguation were discussed by Zobia Rehman *et al*. Also, they elaborated the continual nature of Urdu language and discussed the problems occurred by this nature. When we see the English language, then each letter in an English word has its specific form but when we see Urdu then form of each letter changes its shape as discussed above by an example when appeared in a word w.r.t its position. There are many algorithms as well as stemmers have been proposed to occupy the morphological nature of languages like Arabic language and Urdu language [12].

In many sentences the stop words occur at different events and their semantic significance is no more important in the order they occur. Information retrieval framework gave the idea of stop words for the first time [13]. For Urdu language, there has been a great deal of work done yet, but the elimination process of stop words from Urdu language is still unfocused. A great number of analysts have proposed versatile algorithms for proofing distinguished lists and removal of stop words from many languages. A Support Vector Machine was utilized by Catarina Silva and Bemardete Ribeirot to decide the significance of elimination of stop words on Recall Values from a Text Categorization [14]. A system was proposed by Walaa Medhat et al to create a stop words elimination in an Arabic Online Social Network (OSN) corpora. They at first showed a methodology for the course of action of corpora in Arabic vernacular from OSN and a short time later they reviewed goals with the ultimate objective of SA (Sentiment Analysis). Then they proposed a system at that point which produced the stop words list from corpora [15]. The unpredictability of stop words was demonstrated by Eduard Dragut et al. They proposed the estimation algorithm for the detailing of this issue with regards to Web inquiry interface joining [16]. Naïve Bayesian (NB) were compared by Bassam Al-Shargabi et al., with Sequential Minimal Optimization (SMO) Support Vector Machine (SVM) and J48 were utilized to choose the exactness of each classifier for the portrayal of Arabic language by elimination of stop words. They gauged the exact classifier by using K-fold Percentage split (holdout) and cross approval procedure [17].

Chinese language stop words list were generated by a proposed aggregated automated algorithm by Feng Zou *et al* that is based on statistical and information model. The list that was generated by their proposed algorithm when compared with English stop words list and it was more general when compared with Chinese language words [18]. The Hindi words sense disambiguation and stemming and the removal effect of stop words were investigated by Satyendr Singh and Tanveer Siddiqui [19]. The grouping of characteristic language content was upgraded by Hakan Ayral and Sırma Yavuz and they exhibited a programmed procedure for separating area of stop words. The Bayesian characteristic language classifier was actualized by them, which depended on most extreme a posteriori likelihood estimation of appropriations utilizing bag-of-words model to test the produced stop words and it taken a shot at website pages. The stop words list of English languages was compared with their proposed model [20]. An ideal algorithm was developed by Al-Shalabi et al for the extraction of stop words from archives of Arabic dialect. A Finite State Machine (FSM) was in their view of pseudo code. They made stop-list comprised of more than 1000 words. Al-Shalabi et al tried their algorithms on an arrangement of information browsed the dataset and the Holy Quran. The data set contained the 242 Arabic modified works looked by [21].

III. Proposed Methodology

Normally common grammatical mistakes occur in Urdu language when dealing with stop words. When we are using Urdu syntax, then many people used to confuse کہwithکے, پہ with پےand with those words that have different meanings. In the domain of natural language processing (NLP), a considerable measure of work has been done in areas like, stemming for Urdu language and sentence limit disambiguation. That is the reason we associated the algorithms proposed by [21] on Urdu language. In this article, we are very first who are applying the stop word removal algorithms. As we know that Arabic script is used for composing of Urdu and Arabic language contains an extensive part of it. So that's why, our proposed stop words removal algorithm is similar with [21]. When applying the algorithm, we will propose a productive procedure for Urdu content mining which extricate all sort of stop words from Urdu archives. There are different procedures to peruse that word reference or document. Depicted strategy that have been performed is to traverse the order of lexicon and compare the words in the document unless we find the stop words. Another technique is to take the O(n) in linear search and in the case of time in binary search it will be O(logn).

A. Algorithm Description

We have used deterministic finite automate (DFA) in this study to define our proposed algorithmic methodology for Urdu text mining. A Finite State Machine is used to present the all separated stop words from stop-list. The varying

number of rows were extracted with 38 columns of state tables of DFA in RAM. The rows count varies when used in Urdu Language, and a content word due to its continual nature of occurrence in a document will then become a stop word. The columns which were used are 38 in total count and are the followings: (حروفِ تہجی) ا، ب، پ، ت، ٹ، ث، ج، چ، ح، خ، د، ڈ، ذ، ر، ڑ، ز، ژ، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ک، گ، ل، م، ن، و، ہ، ء، ی، ے۔

In figure 2, we have shown the flow of our algorithm and the algorithm will be as follows:

Initial Stage:

Input: Urdu text

Output: Stop words list from Urdu text.

A. *Check for the word that is first in string;*
B. *If non-Urdu letters found or any kind of special expressions come, then remove those letters and start counting letters again.*
C. *If length of word <= 3,*
   After this the current word will be the stop word and we will then get the word coming next and go-to B.
   Else
   New_state and count will be intilized with 1.
   While new_state >0 and count < word length.
   Assign the intersection value of new_state row to new_state with letter and increment the count of column position from the state table.
   Loop will be ended
D. *If new_state = final state and count > word length*
   Current word = stop word.
   Take the next word and go-to B.

B. Implementation

A constructed DFA will be used to implement our proposed methodology and to see whether our DFA will accept the coming word as a stop word or it will find another one. After this we will convert the DFA into a state table. Our proposed DFA sample will then take the following words: کا، کہ، یہ، تھا، تھی، تو and these are also shown in figure 3.

DFA is also shown in figure 3 alongside and currently we are implementing our proposed algorithm and soon we will publish the results. Due to the script similarity of Urdu and Arabic languages, we expect from our algorithm an efficient performance.

IV. Elaboration of Constructed DFA

Our DFA will be elaborated with the examples shown in below string:۔اسلام علیکم کتابیں پڑھتا ہوں اور لکھتا ہوں

The string will only accept those words that will start with ک and ت and those words will be stop words. Then that words will end with ا، ی، ے and و. Hence, our string will accept the کا and تھا. The words that are remaining from the string will be categorized as content words.

۔پرسوں اسکی امی کیوں گئی تھی

In the given example, the accepted words as the stops words are تو، نہ، ہو، کی and تھی Also, the words remaining will then categorized as content words.

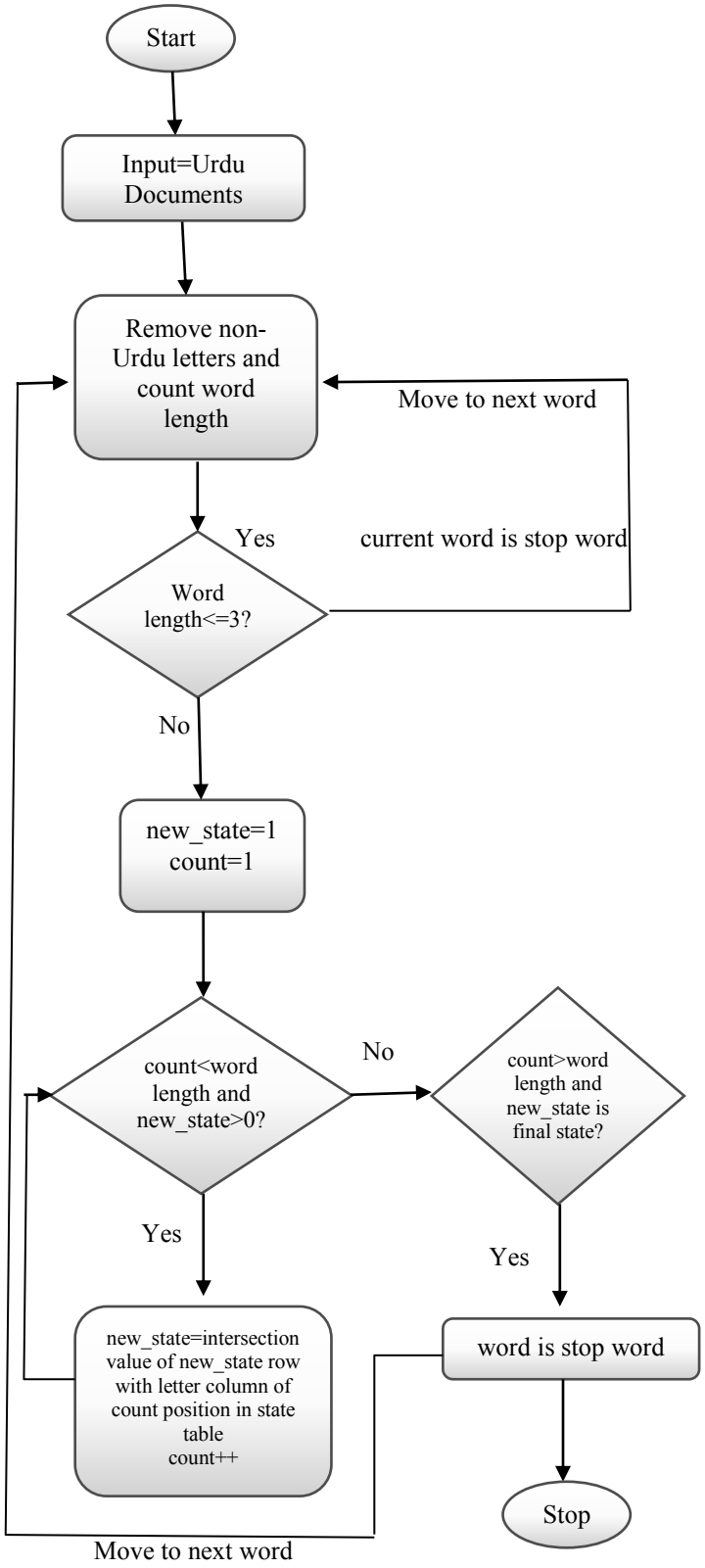

Fig. 2. Flow Chart of Proposed Algorithm

V. CONCLUSION AND FUTURE WORK

In our resultant study, we have proposed an efficient algorithmic approach for elimination cause of stop words in an Urdu text document through text mining technique. In near future, we will try to extend this paper and provide the extended methodology for an optimal and efficient algorithmic approach for the elimination of stop words from an Urdu document. One can also try to do this work by extending our proposed algorithm in an efficient way.

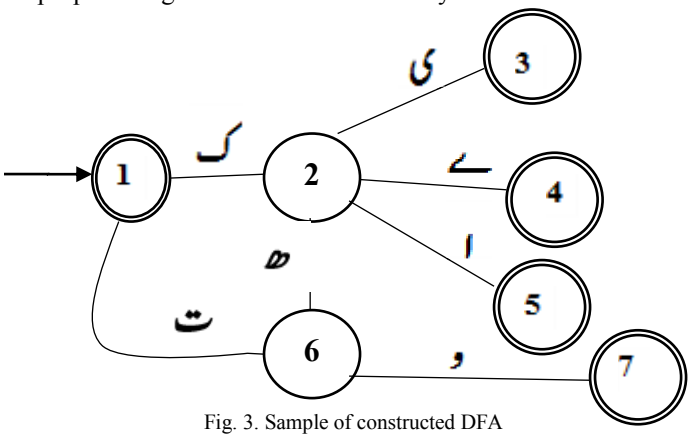

Fig. 3. Sample of constructed DFA

Table II: Sample of Constructed State Table

	ت	ھ	ک	و	ے	ی	ا
1	6		2				
2		6	1		5	4	3
3*							2
4*						2	
5*					2		
6	1	2		7			
7*				6			

REFERENCES

[1]. Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. Vol. 1. No. 1. Cambridge: Cambridge university press, 2008.

[2]. Khan, Sajjad, et al. "Template based affix stemmer for a morphologically rich language." The International Arab Journal of Information Technology 12.2 (2015): 146-54.

[3]. Anwar, Waqas, Xuan Wang, and Xiao-long Wang. "A Survey of Automatic Urdu language processing." Machine Learning and Cybernetics, 2006 International Conference on. IEEE, 2006.

[4]. Hardie, Andrew. "Developing a tagset for automated part-of-speech tagging in Urdu." Corpus Linguistics 2003. 2003.

[5]. Wali, Aamir, and Sarmad Hussain. "Context sensitive shape-substitution in nastaliq writing system: Analysis and formulation." Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. Springer Netherlands, 2007. 53-58.

[6]. Akram, Qurat-ul-Ain, Asma Naseer, and Sarmad Hussain. "Assas-Band, an affix-exception-list based Urdu stemmer." Proceedings of the 7th Workshop on Asian Language Resources. Association for Computational Linguistics, 2009.

[7]. Riaz, Kashif. "Rule-based named entity recognition in Urdu." Proceedings of the 2010 Named Entities Workshop. Association for Computational Linguistics, 2010.

[8]. Larkey, Leah S., Lisa Ballesteros, and Margaret E. Connell. "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[9]. Aqil Burney, Badar Sami, et al. "Urdu Text Summarizer using Sentence Weight Algorithm for Word Processors."

[10]. Puri, Rajeev, R. P. S. Bedi, and Vishal Goyal. "Punjabi Stemmer Using Punjabi WordNet Database." Indian Journal of Science and Technology 8.27 (2015).

[11]. Hussain, Sarmad, Nadir Durrani, and Sana Gul. "Survey of Language Computing in Asia." Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences 2 (2005): 2005.

[12]. Rehman, Zobia, Waqas Anwar, and Usama Ijaz Bajwa. "Challenges in Urdu text tokenization and sentence boundary disambiguation." Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP 2011). 2011.

[13]. Amarasinghe, Kasun, Milos Manic, and Ryan Hruska. "Optimal stop word selection for text mining in critical infrastructure domain." Resilience Week (RWS), 2015. IEEE, 2015.

[14]. Silva, Catarina, and Bemardete Ribeiro. "The importance of stop word removal on recall values in text categorization." Neural Networks, 2003. Proceedings of the International Joint Conference on. Vol. 3. IEEE, 2003.

[15]. Medhat, Walaa, Ahmed H. Yousef, and Hoda Korashy. "Egyptian Dialect Stopword List Generation from Social Network Data." arXiv preprint arXiv:1508.02060 (2015).

[16]. Dragut, Eduard, et al. "Stop word and related problems in web interface integration." Proceedings of the VLDB Endowment 2.1 (2009): 349-360.

[17]. Al-Shargabi, Bassam, Waseem Al-Romimah, and Fekry Olayah. "A comparative study for Arabic text classification algorithms based on stop words elimination." Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications. ACM, 2011.

[18]. Zou, Feng, et al. "Automatic construction of Chinese stop word list." Proceedings of the 5th WSEAS international conference on Applied computer science. 2006.

[19]. Singh, Sushil, and Tanveer J. Siddiqui. "Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation." Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on. IEEE, 2012.

[20]. Ayral, Hakan, and Sırma Yavuz. "An automated domain specific stop word generation method for natural language text classification." Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on. IEEE, 2011.

[21]. Al-Shalabi, Riyadh, et al. "Stop-word removal algorithm for Arabic language." Proceedings of 1st International Conference on Information & Communication Technologies: from Theory to Applications, CTTA'04. 2004.