Stop Words Elimination in Urdu Language using Finite State Automaton

Kamran Shaukat[1*], Muhammad Umair Hassan[1], Dr. Nayyer Masood[2] and Ahmad Bin Shafat[3]

[1*, 3]Department of Information Technology, University of the Punjab
Jhelum Campus, Jhelum, Pakistan

[1]School of Information Science and Engineering, University of Jinan
336 West Road, Jinan, China

[2] Department of Computer Science, Capital University of Science and Technology, Islamabad

kamran@pujc.edu.pk[1*], 20172410007@mail.ujn.edu.cn[1], nayyer@cust.edu.pk[2], bcs.f12.04@pujc.edu.pk[3]

Abstract

Stop words have multiple occurrences in many sentences and have least semantic importance in the context in which they appear. Stop words cover a major volume of documents having very little semantic importance. So they should be removed for better language processing and classification. In this research study, we have designed and proposed an efficient algorithm for the elimination of stop words from Urdu documents. There have been a lot of work in domains like natural language processing (NLP), sentence boundary disambiguation and stemming for Urdu language but we are unaware of any work or methodology proposed for the elimination of stop words from Urdu language. That is why we applied the algorithm proposed by (Al-Shalabi et al., 2004) on Urdu language. As per the best of our knowledge, we are the first to apply any kind of algorithm for Urdu language stop word elimination.

Keywords

Stop word; Urdu; natural language processing; stemming.

1. Introduction

Those words which are extremely common in a document and they carry least semantic value in the document are called stop words (Manning et al., 2008). These words are just used for the grammatical restrictions. In Urdu Language, these words include, but not limited to کے کا، کی، تھا، تھی، تھے، جو،گے،گا،گی،پر،ہر،تو،اور،کہ،تر،ہی،لی،لے،چکی،چکا،چکی،.

Urdu is categorized as the Indo-Aryan branch of Indo-European languages. Its script is written in left-to-right direction and more than 300 million people speak Urdu as their native or secondary language worldwide (Khan et al., 2015; Anwar et al., 2006; Hardie, 2003; and Riaz, 2010). The word “Urdu (اردو)” is a derivative form of Turkish word “ordu” which means “Tent”, “horde” or “Army”. That’s why Urdu is called “Lashkari Zuban (لشکری زبان)” i.e. “The Language of the Army” (Khan et al., 2015). Urdu is spoken in Pakistan as a national language. The twenty-three official languages of India include Urdu language too. Urdu is written in Perso-Arabic and Arabic Script (Khan et al., 2015; and Wali and Sarmad, 2007). Urdu is very close to Hindi but a large portion of it consists of Persian, Turkish, Sanskrit, English, Punjabi and Arabic languages (Khan et al., 2015; Aqil et al., 2012; and Rehman et al., 2011).

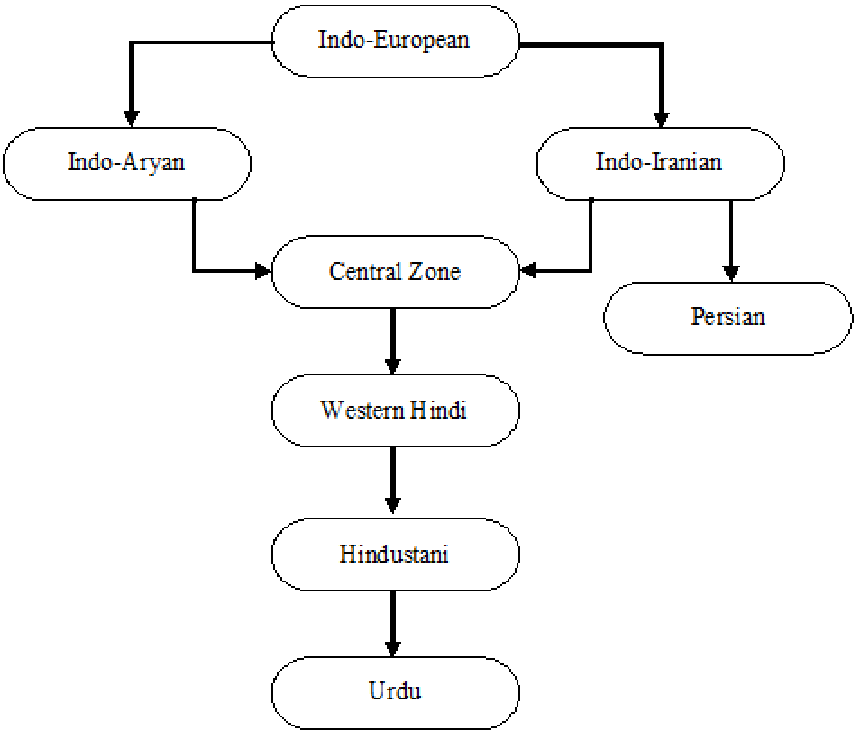

Figure 1: Language Family Tree for Urdu

Urdu is a derivational and inflectional morphological rich language (Akram et al., 2009). Urdu is widely spoken in Bahrain, Afghanistan, Botswana, Bangladesh, Germany, Fiji, India, Guyana, Mauritius, Malawi, Norway, Nepal, Qatar, Oman, South Africa, Saudi Arabia, UAE, Thailand, Zambia and United Kingdom (Hussain et al., 2005). Urdu is a context sensitive language because in an Urdu word, the form and shape of letter differ with respect to its position (start, middle, and end) (Khan et al., 2015). We consider the letter ک for the sake of example. We can see that in the words کیا, بالکل, ابتک letter ک is changing its shape w.r.t its position.

کیا	با لکل	ابتک
Start	**Middle**	**End**

Table 1: Context Sensitivity of Urdu Language

The contribution of this paper is as follows: First we will shortly elaborate different text mining techniques like stemming, stop word elimination and tokenization etc. We will overview the work done in these domains. Then we will explain the different methodologies and algorithms proposed for the elimination of stop words in different languages. Then we will propose an algorithm for removal of stop words in Urdu language.

2. Related Work

Text mining is the process of generating useful information from text data (Amarasinghe et al., 2015). There are multiple methods and techniques for text mining in different languages. Larkey et al., 2002, proposed a statistical stemmer for Arabic retrieval based on co-occurrence. They compared their light and statistical stemmers with a morphological analyzer. For cross-language retrieval, their stemmer performed more efficiently (Larkey et al., 2002). Puri et al., 2015, proposed a Punjabi stemmer for stemming all Punjabi words. They used extended stripping rules and revised suffix removal approach for their stemmer. To find matched suffixes, their algorithm used regular expressions (Puri et al., 2015).

Akram *et al.* proposed Assas-Band, an Urdu stemmer which removed prefix and postfix. Then the stemmer adds letters to make the surface form of the stem. Their stemmer used affix-based exception lists for stemming (Akram et al., 2009). Khan *et al* proposed a template based stemmer to perform stemming for a morphological rich language. Their proposed light stemmer removed all kind of affixes (prefix, postfix and infix). Their stemmer gave 96.08% recall, 89.05% precision and 92.49% FI-Measure (Khan et al., 2015). Zobia Rehman *et al* discussed problems occurred during sentence boundary disambiguation and Urdu text tokenization. They discussed the continuous nature of Urdu language and problems faced due to this nature. In English language, each letter in a word has its own unique form but in Urdu, each letter changes its form while appearing in a word with respect to its position. Zobia *et al* discussed that many stemmers and algorithms had been proposed to overcome this nature of morphological languages like Urdu and Arabic. They have also discussed the ambiguous sentence boundary problem of Urdu (Rehman et al., 2011).

Stop words have multiple occurrences in many sentences and have no semantic importance in the context in which they appear. The concept of stop words was proposed in IR (Information Retrieval) systems first (Amarasinghe et al., 2015). There have been a lot of work for Urdu Language in the domain of Information Retrieval but removal of stop words is still unfocused. Many researchers have proposed different algorithms for identification, list generation and removal of stop words for many languages. Catarina Silva and Bemardete Ribeirot used Support Vector Machine to determine the importance of removal of stop words in Text Categorization on Recall Values (Silva and Bemardete, 2003). Walaa Medhat *et al* proposed a methodology to generate a stop words list of from Arabic OSN (Online Social Network) corpora. They first presented a methodology for the preparation of corpora in Arabic language from OSN (Online Social Network) and then they reviewed sites for the purpose of SA (Sentiment Analysis). Then they proposed a methodology to generate a stop word list from prepared corpora (Medhat et al., 2015). Dragut *et al* proved the complexity of stop word problem and proposed an approximation algorithm for the formulation of this problem in the context of Web query interface integration. Then they have studied the effect of words like AND and OR on the establishment of semantic relation between different labels (Dragut et al., 2009). Al-Shargabi *et al* compared the Naïve Bayesian (NB), SVM (Support Vector Machine) with SMO Sequential Minimal Optimization and J48 to determine the accuracy of each classifier for the classification of Arabic text based on the eliminating stop words. They measured the accurate classifier by using K-fold cross validation and Percentage split (holdout) method. They proved that Sequential Minimal Optimization is the most accurate classifier with minimum error rate. And SMO model can be built in much less time as compared to Naïve Bayesian and J48 classifier (Al-Shargabi et al., 2011).

Zou *et al* proposed an automated aggregated algorithm based on information and statistical model for generation of Chinese language stop words list. Their results showed that their generated stop words list was comparable with English language stop words list and their generated list is more general than other Chinese language stop word lists (Zou et al., 2006). Satyendr Singh and Tanveer Siddiqui investigated the effect of removal of stop words and stemming on the sense disambiguation in Hindi words. They evaluated the impact of stemming and stop words elimination on Hindi WDS on Lesk’s algorithm. They examined the best performance when both elimination of stop words and stemming were involved in the case. They noticed that the context vector had increased number of content words due to stop word elimination and overlapping of reduced morphological variants of the words with same stem was increased due to the stemming effect (Singh and Tanveer, 2012). To enhance the classification of natural language content, Hakan Ayral and Sırma Yavuz presented an automatic methodology for extracting domain specific stop words. They implemented Bayesian natural language classifier which was based on maximum a posteriori probability estimation of keyword distributions using bag-of-words model to test the generated stop words and it worked on web pages. They compared their model generated stop words list with general English language stop word list (Ayral and Sirma, 2011). Al-Shalabi *et al* designed an optimal algorithm for the extraction of stop words from Arabic language documents. Their algorithm based upon a Finite State Machine (FSM). Their created stop-list consisted of more than 1000 words. Al-Shalabi *et al* tested their algorithm on a set of data chosen from the Holy Quran and another dataset which contained 242 Arabic abstracts chosen from Proceedings of Saudi Arabian National Computer Conferences and it gave 98% accurate results (Al-Shalabi et al., 2004).

3. Methodology

In Urdu Language, some grammatical mistakes are very common while dealing with stop words. Some people confuse کہ with کے , پہ with پےand other rhyming words but having different meanings. There have been a lot of work in domains like sentiment analysis, sentence boundary disambiguation and stemming for Urdu language but we are unaware of any work or methodology proposed for the elimination of stop words from Urdu language. That is why we applied the algorithm proposed by (Al-Shalabi et al., 2004) on Urdu language. As per the best of our knowledge, we are first applying any kind of algorithm for Urdu language stop word elimination. Since Urdu is written in Arabic script and a large portion of it consists of Arabic that is why our stop word removal algorithm is very close to (Al-Shalabi et al., 2004). After applying this algorithm, we will propose an efficient methodology for Urdu text mining which extract all kind of domain specific stop words from Urdu documents. Many algorithms or methodologies proposed for the elimination of stop words, use a prepared dictionary or list which consists of language specific stop words. There are multiple techniques to read that dictionary or file. Described method is to traverse the file or dictionary, perform comparison for each word unless stop words are identified or we reach the end of file. The other way is to use binary search. This type of methodologies takes O(logn) time in binary search and O(n) in linear search.

3.1. Proposed Algorithm

In this research study, we have applied an algorithm to enhance the stop word elimination process from Urdu text using Deterministic Finite Automata. We separated all stop words in stop-list, and presented them in Finite State Machine. Then we implemented the DFA in RAM as state table which was consists of 38 columns and varying number of rows (states). Number of rows vary because in Urdu Language, sometimes, a content word becomes a stop word due to its frequent occurrence in a document. 38 columns consist of following Urdu alphabetical letters (حروف ِ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۔

The flow chart for our proposed algorithm is shown in Figure 2. The algorithm for stop word elimination is as follows:

Initial Stage: 0

Input: Urdu text

Output: List of stop words in the given Urdu text.

```
A.  Go to the first word in the text;
B.  If there are any non-Urdu letters or special characters, remove them and count the word length
C.  If word length is less than or equal to 3,
    Then current word is a stop word,
    get the next word and go to B.
    Else
    Initialize count and new_state with 1
    While count< word length and new_state >0
    new_state=intersection value of new_state row with letter column of count position in state table increment count
    Loop ended
D.  If count is greater than word length and new_state is final state in state table
    Current word is stop word
    Get the next word and go to B.
```

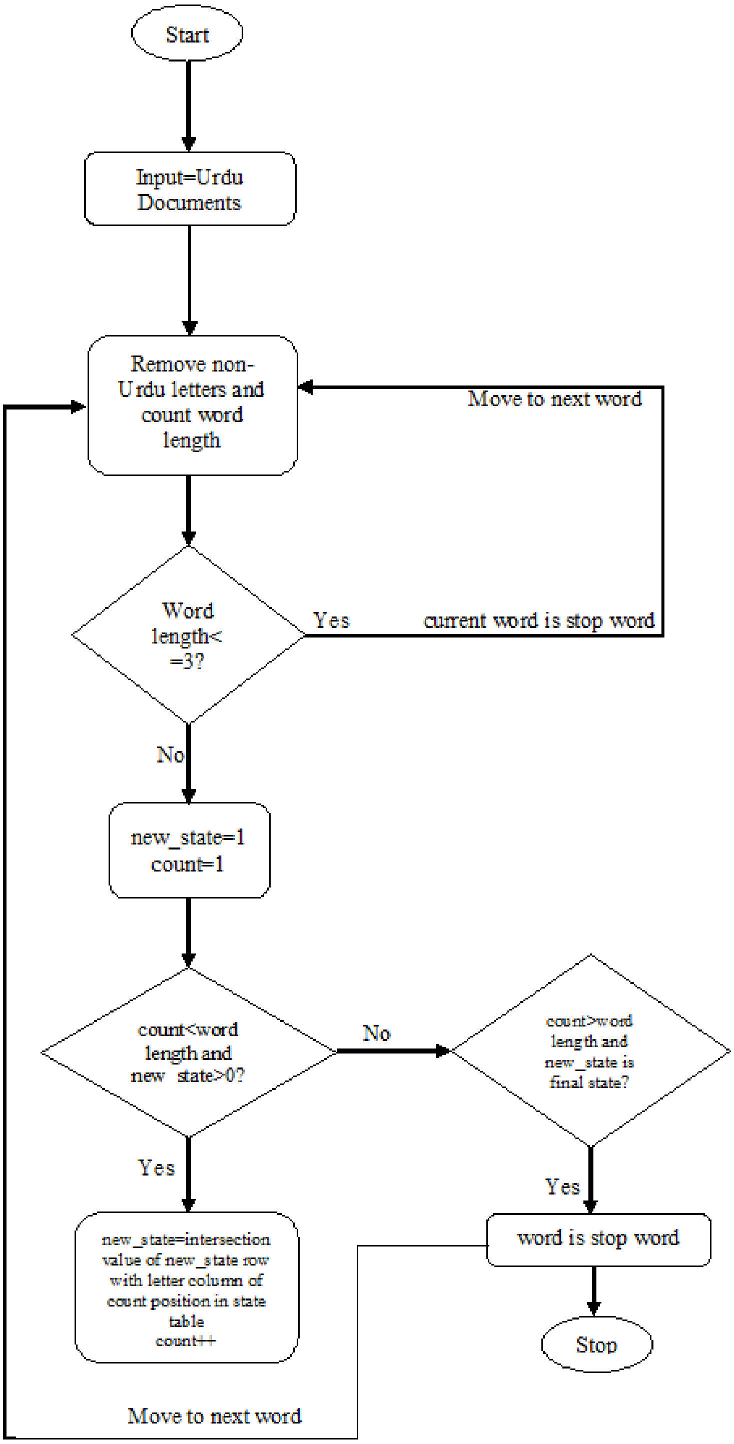
Start
Input=Urdu Documents
Remove non-Urdu letters and count word length
Move to next word
Word length<=3?
Yes
current word is stop word
No
new_state=1
count=1
count<word length and new state>0?
No
count>word length and new_state is final state?
Yes
new_state=intersection value of new_state row with letter column of count position in state table
count++
Yes
word is stop word
Stop
Move to next word

Figure 2: Flow chart of proposed method for Stop Words removal

3.2. Implementation of Algorithm

To implement our proposed algorithm, we will use a constructed DFA to see that either it accepts a word as a stop word or not. Then we will convert our deterministic finite automata to a state table. A sample of our DFA which takes the following Urdu words: کا، کے،کی، تھا، تھے،تھی، تو and کھو is shown below in Figure 3.

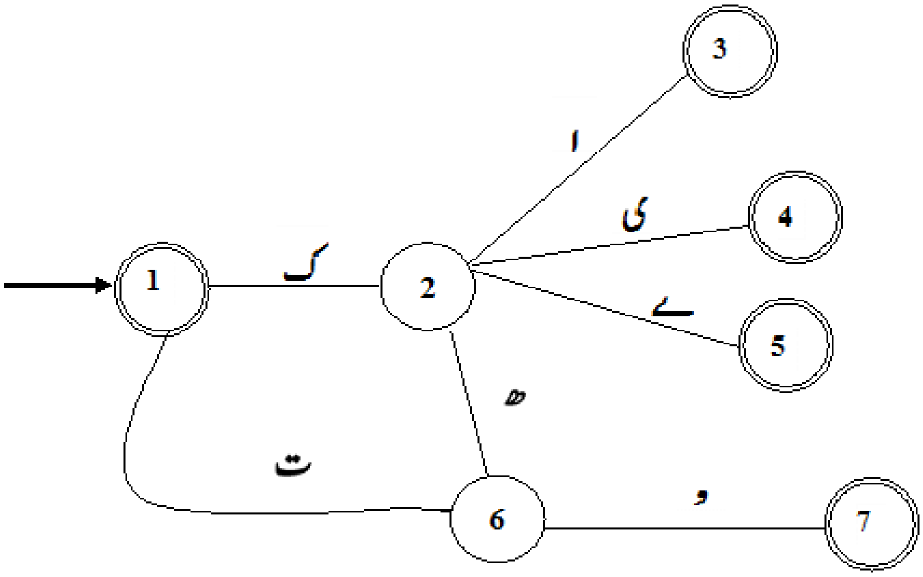

Figure 3: Proposed Deterministic Finite Automaton

Different Urdu alphabetical letters are presented in table columns and rows present the state numbers. Rows with asterisk means that these states are final states. The value in the table present the next state in the deterministic finite automata for current word. Translated state table, which represent the DFA is also shown alongside.

We are currently working on the Implementations of the proposed algorithm and awaiting results. But due to script similarity of Arabic and Urdu language, we are hopeful that our proposed algorithm will perform efficiently.

4. Constructed DFA and Description

Now we will elaborate the sample of our DFA with the help of following examples:

احمد اس کا دوست تھا۔

When we run this sentence on this sample, it will only accept those stop words starting with ک and ت. And ending with ے، ی، اand و. So it will accept کا and تھا. And remaining words are separated as content words.

گھڑی تو اس کے ابو کی کھو گئی تھی۔

Now the accepted stop words will be تو، کھو، کیand تھی. Other words will be categorized as content words.

This DFA only accepts the stop words and it can be applied to other similar context languages for further enhancement of work.

	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**			

Table 2: Constructed Sample State

5. Conclusion

This paper discussed the problems of removing stop words from Urdu documents only. We are currently working to optimize our algorithm to apply it to other contextual languages and the accuracy has been tested using lucene and this work is novel and up-to date. According to our knowledge, we are the first to apply any sort of algorithm for stop word removal from Urdu language. For future work, one can try to enhance this algorithm and propose even more optimal methodology for removal of stop words from Urdu language.

6. References

Akram, Q.U.A., Naseer, A. and Hussain, S., 2009, August. Assas-Band, an affix-exception-list based Urdu stemmer. In *Proceedings of the 7th workshop on Asian language resources* (pp. 40-46). Association for Computational Linguistics.

Al-Shalabi, R., Kanaan, G., Jaam, J.M., Hasnah, A. and Hilat, E., 2004, April. Stop-word removal algorithm for Arabic language. In *Proceedings of 1st International Conference on Information and Communication Technologies: From Theory to Applications, CTTA* (Vol. 4, pp. 19-23).

Al-Shargabi, B., Al-Romimah, W. and Olayah, F., 2011, April. A comparative study for Arabic text classification algorithms based on stop words elimination. In *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications* (p. 11). ACM.

Amarasinghe, K., Manic, M. and Hruska, R., 2015, August. Optimal stop word selection for text mining in critical infrastructure domain. In *Resilience Week (RWS), 2015* (pp. 1-6). IEEE.

Anwar, W., Wang, X. and Wang, X.L., 2006, August. A Survey of Automatic Urdu language processing. In *Machine Learning and Cybernetics, 2006 International Conference on* (pp. 4489-4494). IEEE.

Ayral, H. and Yavuz, S., 2011, June. An automated domain specific stop word generation method for natural language text classification. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on* (pp. 500-503). IEEE.

Burney, A., Sami, B., Mahmood, N., Abbas, Z. and Rizwan, K., 2012. Urdu Text Summarizer using Sentence Weight Algorithm for Word Processors. *International Journal of Computer Applications*, *46*(19).

Dragut, E., Fang, F., Sistla, P., Yu, C. and Meng, W., 2009. Stop word and related problems in web interface integration. *Proceedings of the VLDB Endowment*, *2*(1), pp.349-360.

Hardie, A., 2003. Developing a tagset for automated part-of-speech tagging in Urdu. In *Corpus Linguistics 2003*.

Hussain, S., Durrani, N. and Gul, S., 2005. Survey of language computing in Asia. *Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences*, *2*, p.2005.

Khan, S., Anwar, W., Bajwa, U. and Wang, X., 2015. Template Based Affix Stemmer for a Morphologically Rich Language. *International Arab Journal of Information Technology (IAJIT)*, *12*(2).

Larkey, L.S., Ballesteros, L. and Connell, M.E., 2002, August. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings* *of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275-282). ACM.

Manning, C.D., 2008. Prabhakar Raghavan, and Hinrich Schutze. *Introduction to information retrieval*.

Medhat, W., Yousef, A.H. and Korashy, H., 2015. Egyptian Dialect Stopword List Generation from Social Network Data. *arXiv preprint arXiv:1508.02060*.

Puri, R., Bedi, R.P.S. and Goyal, V., 2015. Punjabi stemmer using punjabi wordnet database. *Indian Journal of Science and Technology*, *8*(27).

Rehman, Z., Anwar, W. and Bajwa, U.I., 2011, November. Challenges in Urdu text tokenization and sentence boundary disambiguation. In *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP* (Vol. 2011, pp. 40-45).

Riaz, K., 2010, July. Rule-based named entity recognition in Urdu. In *Proceedings of the 2010 named entities workshop*(pp. 126-135). Association for Computational Linguistics.

Silva, C. and Ribeiro, B., 2003, July. The importance of stop word removal on recall values in text categorization. In Neural Networks, 2003. Proceedings of the International Joint Conference on (Vol. 3, pp. 1661-1666). IEEE.

Singh, S. and Siddiqui, T.J., 2012, March. Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation. In *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on* (pp. 1-5). IEEE.

Wali, A. and Hussain, S., 2007. Context sensitive shape-substitution in nastaliq writing system: Analysis and formulation. *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, pp.53-58.

Zou, F., Wang, F.L., Deng, X., Han, S. and Wang, L.S., 2006, April. Automatic construction of chinese stop word list. In *Proceedings of the 5th WSEAS international conference on Applied computer science* (pp. 1010-1015).