World Applied Sciences Journal 7 (Special Issue of Computer & IT): 199-205, 2009
ISSN 1818.4952

Persian Document Summarization by Parsumist

Mehrnoush Shamsfard, Tara Akhavan and Mona Erfani Joorabchi

NLP Research Laboratory, Department of Electrical and Computer Engineering,
Shahid Beheshti University, Tehran, Iran

Abstract: The rapid growth of online information services has created the problem of information explosion. Automatic text summarization techniques are essential for dealing with this problem. The process of compacting a source document to reduce its complexity and length while retaining its most important contents is called text summarization. This paper introduces Parsumist-a text summarization system for Persian documents. It exploits a combination of statistical, semantic and heuristic-improved methods. It can generate generic or topic/query-driven extracts summaries for single-or multiple Persian documents. In this paper, we first review the related work in this field, especially for Persian text summarization. We then present the architecture of Parsumist, its components and features. The last section evaluates the system and compares it to other systems that exist.

Key words: Automatic text summarization . multi document summarization . extraction . lexical chains . persian . semantic relations

INTRODUCTION

Nowadays there is a vast amount of textual information on the web. It is too difficult for users to read and locate their needs in such a bulky information repository. Therefore, a summarization system would be helpful to allow users (1) to find the resources they need more rapidly and (2) to access the most important parts of the texts.

In other words, text summarization is the process of extracting the most important parts of information from source document(s) to produce a compact version for a particular user or task.

Automatic text summarization can be used in various areas of applications such as intelligent tutoring systems, telecommunication industry, information extraction and text mining, question answering, news broadcasting and word processing tools. Researchers are also investigating the application of this technology to a variety of inputs such as single vs. multi documents, mono vs. multi lingual texts, text vs. speech inputs, single vs. multi media and online vs. offline summarization.

In this paper, we introduce Parsumist, an automatic summarization system developed for summarizing single-and multiple documents in the Persian language. Our approach to multi-document summarization is an extension of our single-document approach. It exploits a combination of statistical, semantic and heuristic-improved methods to extract summaries from single documents and then omitting redundancies, organizes the selected sentences intended for the final summary. It performs in both generic-and query-oriented modes and extracts the most important sentences from the text.

The remainder of the paper is organized as follows: Section 2 discusses related works. Section 3 introduces Parsumist and Section 4 shows the experimental results. Finally, the Conclusion discusses current-and future efforts being made to improve the summaries generated.

RELATED WORKS

There are various approaches to summarization, some of which have been extant for about 50 years. Text summarization approaches can be categorized in different ways according to the various features. For example, according to Hovy and Lin [1] the features are related to input, purpose and output of systems and result in different types of summary such as extracts (selecting some pieces of original text) vs. abstracts (paraphrasing and generating a shorter text); indicative (keywords indicating topics) vs. informative (content laden); generic (author's perspective) vs. query-oriented (user-specific); background vs. just-the-news; single-document vs. multi-document and neutral vs. evaluative.

Corresponding Author: Dr. Mehrnoush Shamsfard, Department of Electrical and Computer Engineering, Shahid Beheshti University, Evin, 1983963113, Tehran, Iran

Extracting summaries, which is the focus of this paper, can be done between surface-and deep levels using various approaches such as shallow text understanding [1-3], statistical and corpus based [4], discourse structure based [5] and knowledge based ones [6, 7].

Classical methods use simple statistical parameters such as frequency of keywords and title words, length of sentence, or an occurrence of cue words to find salient sentences. Lexical chain [5] is another popular technique in which a chain of related terms is built up and a scoring method evaluates each chain. The stronger chain is more likely to appear in the summary.

In graph-based methods, a graph is constructed in which the nodes represent sentences and edges show the similarity between sentences. To determine the similarity between two sentences, systems could focus on the overlaps of words, synonyms, verb/argument structures, stems, etc. They could calculate the similarities among sentences by metrics such as the cosine similarity metric.

Nowadays, exploiting new methods and techniques to improve the performance of summarization systems is shifting from shallow processing of texts towards deeper analysis and greater attention to semantic features. In the new trend that this shift represents, linguistically motivated natural language processing techniques, including semantic processing, discourse analysis, text understanding, automated reasoning, supervised-or unsupervised machine learning, ontology based techniques and the like play the major roles in creating better summaries. For example, Summarist [1] has been developed to extract sentences from single documents, but now some work is underway both to extend the extract-based capabilities of SUMMARIST and to build up the large knowledge collection required for inference-based abstraction.

Multi-document summarization (MDS), which is very popular these days, is the extension of single document summarization to collections of related documents [8]. The main focus of MDS is summarizing texts while removing redundancy and taking into account the similarities and differences in the information content of different sources.

There are two major groups of approaches to handle multiple documents. The first group uses the usual methods of single-document summarization treating each member of the document set as a single document to generate its summary of desired length. Then combining all of the summaries together it again summarizes them to produce the final summary.

The second group of approaches is specifically designed for multiple documents. In these approaches, the sentences of the summary are extracted from all of the documents together using graphs and clustering methods. This approach is more challenging, intelligent and complicated. An example of this group is the SUMMONS system [9] which extracts and combines information from multiple sources and passes it on to a language generation component to produce the final summary

There are some midway methods too which behave as single-document methods on inputs and then introduce new techniques to merge the summaries [10].

Persian text summarization: There are many summarization methods and systems available for languages such as English. Although some of them claim to be language-independent, they need at least language resources to work with. The lack or shortage of these resources such as training and test data, lexical ontologies or semantic lexicons, lists of stop words and cue-words and even fundamental language processing tools such as reliable tokenizers, stemmers and parsers all make text summarization a hard task for languages such as Persian with less resources. In contrast to English summarization systems, summarization of single-and multiple documents written in Persian is a new, ongoing research effort.

The oldest work on Persian text summarization is FarsiSum [11]. It is an HTTP client/server application programmed in Perl based on Swesum [12], a summarizer for the Swedish language. FarsiSum extracts data from single documents with the main body of language-independent modules implemented in SweSum. In FarsiSum, the Persian stop-list has been added in Unicode format and the interface modules is adapted to accept Persian texts.

The second work is a single document Persian text extractor based on lexical chains and graph-based methods [13]. This System uses 5 measures: namely similarity to other sentences, similarity to user's query, similarity to the title and the number of common words and cue words to score a sentence. Some specific Persian resources to prepare the chains and graphs are used in its scoring module.

Honarpisheh and his colleagues [14] have developed a multi-document multi-lingual text summarizer based on singular value decomposition and hierarchical clustering. Their approach relies on only two resources for any language: a word segmentation system and a dictionary of words in conjunction with their document frequencies. The summarizer initially receives a collection of related documents and transforms them into a matrix; it then applies singular value decomposition to the resulting matrix. Using a binary hierarchical clustering algorithm, it then chooses the most important sentences of the most important clusters to create the summary.

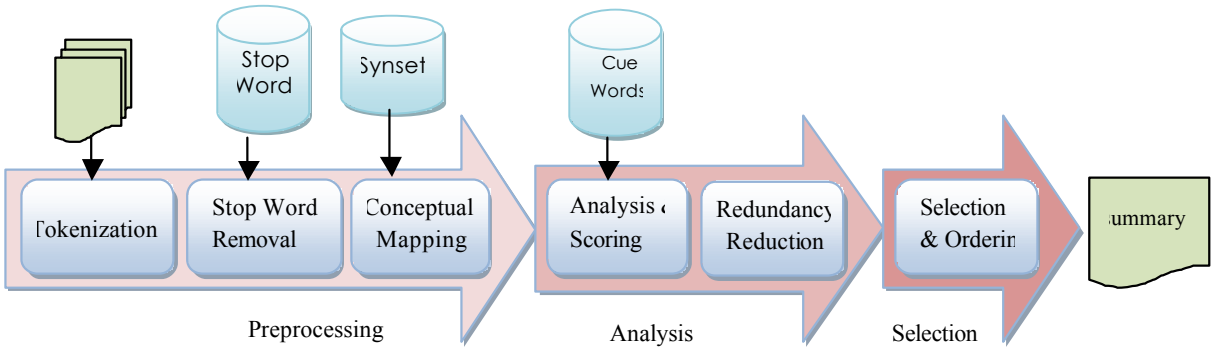

Fig. 1: Architecture of PARSUMIST

Our system is similar to the system introduced by the authors of reference [13] as it uses lexical chains as well, but we have improved their work by using semantic features and representing a conceptual meaning of the text using synonym sets, applying redundancy checking, smoothing the summary for coherence and making it applicable for both single and multi-documents. We have also developed some resources, tools and heuristics for the Persian language. Our system can create a general summary or prepare a query-based summary.

AN INTRODUCTION TO PARSUMIST

The architecture of Parsumist is shown in Fig. 1.

As can be seen, the system consists of three main parts: preprocessing, analysis and selection. The main resources used in this system are stop-words, cue words and synonym sets of Persian words. On receiving a document, Parsumist refines it by deleting the stop words and rewriting the others with their corresponding concepts (if available). It then scores each sentence according to lexical chains and selects the most important ones for inclusion in the summary. It also checks for redundancy to avoid repeating similar sentences among the sentences selected. Finally, the system exploits some heuristics and attempts to make the summary smooth and coherent.

In the following subsections we describe the main modules of Parsumist in more detail.

The preprocessing section receives the document(s), the compression ratio and the possible query from the user and performs the following tasks:

- Tokenization: The first step in processing texts is recognizing the boundaries of text constituents, such as sentences, phrases and words. This process is called tokenization. Although there are some known tokenization algorithms, there are some more problems when it comes to tokenizing Persian texts. Space which is an orthographic word boundary delimiter in English is not the precise and deterministic boundary of distinct words in Persian. Compound verbs with long distance dependencies, omitting Ezafe in Ezafe construction, a variety of prescribed forms of Persian writing and free word order-all characteristics of the Persian language-are some of the major problems in preprocessing Persian texts [15].
- Stop word removal: We have developed a list of stop words for Persian. It is an improved version of the list created by [13]. At this stage we take a copy from the original document and remove all stop words from it. The original copy is retained for extraction of final sentences.
- Conceptual mapping: The last part of preprocessing is conceptual mapping. At this stage, all words and phrases are replaced by their corresponding concepts, if available in our sets of synonyms. The repository of synonym sets should be replaced by a lexical ontology in further extensions or modifications but as there is no lexical ontology for Persian yet, we have created a small limited one containing only the synonymy relationships for the purpose of testing our algorithm. Each synonym set has a representative which is entered into the text whenever any of the other members of that set occurs. This helps in redundancy checking in the steps that follow.

Analysis and scoring sentences: At this stage, we assign a significance score to each isolated sentence. To do this certain statistical parameters are measured according to some linguistic resources such as the lists of stop-words and cue words and also some heuristic rules. A few of the parameters for which a search is carried out in each sentence are:

- The number of main words, title words and query words: the higher numbers indicate the more important sentences.

- The length of the sentence: Shorter sentences are more likely to appear in the summary.
- Proper nouns: The importance of proper nouns depends on the type of document. Their importance might be higher or lower in news, stories, or scientific articles.
- English words and phrases: In Persian texts, especially scientific articles, it is quite common to see English translation of keywords at their first occurrence. Thus, an English word appearing in a sentence acquires importance if the sentence is describing a scientific phenomenon or introducing a new term.
- Quotation marks: Quotation marks are used to quote an important speech and thereby enhance the significance of the sentence.
- Pronouns: If the summary includes a sentence containing a pronoun, it should include the sentence containing the reference of the pronoun too. To solve this problem, we should either delete from the summary the sentence with the pronoun or include the sentence that containing its reference too. To use the second solution we have to either resolve the reference (which is a complicated task) or believe that in 80% of cases the reference for a pronoun appears in the previous sentence and hence increase the significance of the previous sentence if the score of current sentence is high. In our system, we reduce the significance score of a sentence on finding it contains a pronoun (but we do not delete it).
- % sign-The percentage sign is mostly used to qualify some results and usually is an important piece of information. It increases the sentence score.
- Parenthetical-or descriptive sentences-Embedded sentences and phrases-whether surrounded by parentheses or not-are usually used to add to the description of a concept and therefore may be deleted from a summary.
- Referential phrases-Phrases that refer to some other parts of a document (e.g. in the last section, in the previous figure, the next paragraph etc.) should not appear in the summary unaltered. They should be changed and/or gain lower scores.
- Punctuation marks-Different sentences concatenated by punctuation marks (e.g. he said: "come here.") should be attached together whether included the summary or thrown out of it.

Considering the above parameters, the significance score of each sentence is calculated as a weighted summation using the following formula:

$$W(s_i) = \sum_j c_j p_{ij} \quad (1)$$

In this formula, $W(s_i)$ is the total score of sentence S_i, C_j is the coefficient (weight) of the jth parameter and p_{ij} is the value of the jth parameter for the ith sentence. Although the weight of each parameter can be calculated by means of machine learning techniques, in our test, we assumed all of the weights to be equal to 1.

After calculating the significance score of each isolated sentence, similarity among the sentences has to be computed. Parsumist generates an undirected graph in which the nodes are the sentences and the edges connect similar sentences together. The similarity value is the weight assigned to each edge. The similarity is measured by the number of common-or related words due to the synonym sets and lexical chains. The system considers two sentences similar, if the similarity between them exceeds a specific threshold.

For generating chains, the system uses a greedy method. Thereafter, it scores each chain and thus each edge of the graph. To find related words in a pair of sentences we considered various relations such as equivalence, synonymy, hyperonymy and hyponymy which are arranged in decreasing order of weight. For each sentence, we not only calculate its similarity with other sentences but also compute its similarity with the title and with the user's query (keywords). The most important sentence is the one with the most similarities to other sentences and to the title and keywords and gains the most significance score.

Selection and redundancy reduction: In this step, the final representation of the summary is generated. This means that the set of sentences to be included in the summary and their proper order should be determined. To reduce redundancy in the sentences chosen for inclusion in the summary, we have a three-step process.

(a) Begin with an empty summary.
(b) As long as the summary length is shorter than that desired, choose the sentence with highest score and least resemblance to the previous sentences that have been already selected and include the chosen sentence in the summary. (To do this, we calculate the resemblance of the current sentence with each of the selected sentences, if the resemblance of the important words of the two is more than a specified threshold then the sentence with the greater score is selected and the score of the other sentence is reduced by some percentage.)
(c) Continue thus until the desired summary length is reached.

Multi-document summarization: The multi-document summarization we want to generate is the summary of two or more input documents. For this purpose, we tested two approaches: (1) we concatenated all individual input documents into a single document and then applied our single document summarizer on it. (2) We applied our single document summarizer on each of the input documents separately and then concatenating the results into a new single input. Thereafter with some minor refinements, we applied the summarization method again on the new input to eliminate redundancies and extract the final summary. Our experimental results show that there is no noticeable difference between these two approaches; this could be predicted as we do not consider the structure of documents in our summarization method.

To eliminate redundancies in Parsumist, we applied to multi-document summarization the same algorithms and techniques that we used for single-document summarization with minor differences. In general, for redundancy elimination in the multi-document mode we call the redundancy removal module twice. At first, we generate a summary for each input file so that the redundancies among the sentences of each text are removed. Subsequently, when all the summaries are assembled together redundancy among the summaries of all the texts is checked. We thus minimize redundancy in the final summary.

Smoothing the final summary is another task to be done at this juncture. The sentences extracted from different sources should be interleaved together so that the summary built has a logical temporal trend. Similar sentences in the sources can help in locating common time points in the documents and thus arranging the extracted sentences in a logical order.

DISCUSSION

This section discusses the strengths and weaknesses of Parsumist in comparison to other available Persian summarization systems. The next section presents the experimental results confirming these claims.

- Parsumist is more powerful than corresponding systems in handling coherence of extracts. It assigns penalty weights to sentences containing anaphora and stigma words. The sentences containing these signs should be either selected with their coherent sentences or deleted from the selected sentences. For example, if there is a bulleted list, Parsumist either ignores the list or selects at least one sentence from each bullet (or number) of the list. This feature is very weak in other systems.
- Parsumist eliminates redundancies in both single-document-and multi-document summarization. Redundancy checking is done at sentence level, morphological level and word-semantic level. In short, sentences which are the same-or similar (that is, the same except for synonyms or morphologically variant words) are counted as one. None of the other available systems has this feature.
- Parsumist retains the temporal order of events extracted from different resources better than [14], which is a statistical Persian multi-document summarizer. It also works on single documents far better than [14].
- In developing Parsumist, we have created some language-specific resources for the Persian language such as the list of stop words, cue words and stigma words, a database of synonym words and a set of documents with their human summaries. These resources can be re-used in future systems.
- Parsumist, like other available systems, does not handle gaps in the extracted summary. This is the major limitation of the system which should be dealt with in future studies.
- In the next section, we will see that due to the above strengths of Parsumist over other systems for Persian summarization, experimental results yield better performance according to human evaluations.

EXPERIMENTAL RESULTS

We evaluated our single document-and multi-document summarizers separately. As we had access to the results of two previous single document summarizers (FarsiSum and the summarizer by Karimi and Shamsfard), we tested our work by comparing both with these previous systems and with the gold standard summaries created by humans.

On the multi-document side, we tested the system by comparing the summaries with those created by humans. As there is no tool available to execute the comparison for Persian, we did it manually with the help of some human reviewers.

To create a gold standard summary from human summaries, we gathered some documents from different domains and genres of different lengths ranging from short news in a few sentences to short stories comprising a few hundred sentences. We then selected a set of these summaries and asked a group of more than 20 students in the Computer Engineering field to summarize them with different compression ratios. By this process we gathered at least six human

summaries for each document. The sentences in each human summary were ranked according to their importance by the person preparing the summary; therefore, for each document we could assemble a group of summaries comprising human summaries of different lengths (according to different compression ratios) created by an individual person. This way we generated an excellent set of human summaries with which to test Persian summarization for the first time. Finally, we chose the sentences with the highest frequency in human summaries for inclusion in the gold standard summary. The number of sentences chosen in the gold standard summary depends on the document's length and compression ratio.

To test our system, we performed two types of tests (1) comparison with the gold standard summary (2) comparison with results of other systems.

For the first type, we compared our results with the gold standard created by compression ratio of 30%. Figure 2 shows the precision-recall diagram for the related set of documents. As can be seen, the average precision and recall are both about 65%. Good results including the best precision (85%) and recall (80%) were obtained for scientific reports and articles and the worst cases (precision=37% and recall=42%) were obtained for news articles. This means that Parsumist performs better in scientific-and general documents rather than news articles, although in tests that follow we will see that even in news articles Parsumist performs better than other Persian summarizers.

In the second method, we prepared 5 different summaries for each document, containing 2 human summaries and 3 system summaries from 3 different Persian automatic summarizers available: FarsiSum [11], Karimi and Shamsfard, [13] and Parsumist.

We asked 10 different persons to read the document and 5 summaries related to it and then rank the summaries from the best to the worst. We chose two random document sets with good (scientific) and bad

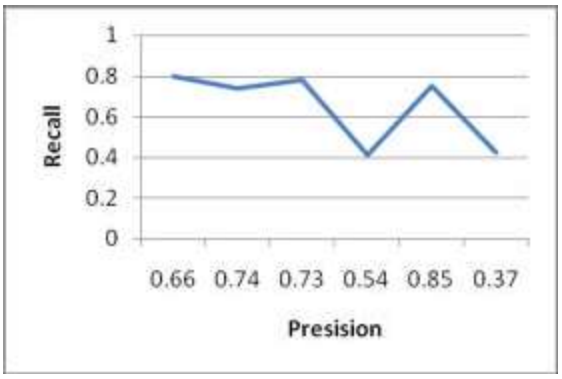

Fig. 2: Test results for parsumist

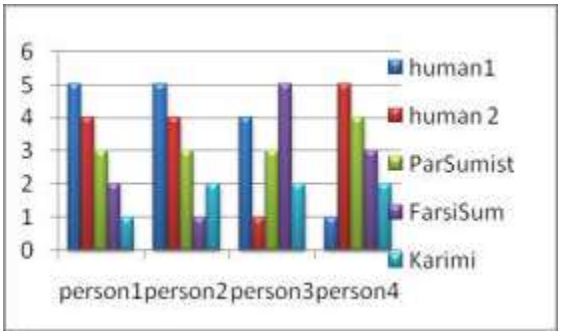

Fig. 3: The results for document set A (good summaries)

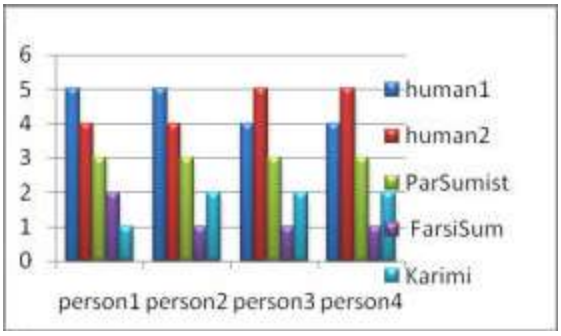

Fig. 4: The results for document set B (bad summaries)

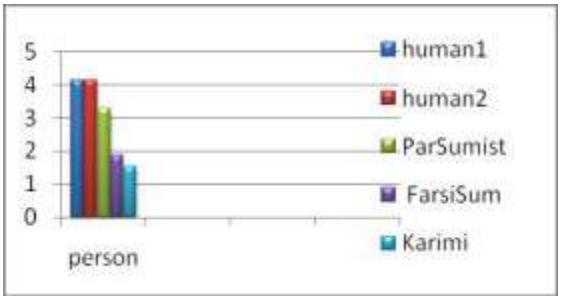

Fig. 5: Average ranking

(news) summaries and present their results in Fig. 3 and 4, respectively. For each document, we show the results of ranking outputs by four reviewers. The results of this method are very similar to those of the first method. It can be seen that in almost all cases the summaries created by Parsumist are better than any summary created by other Persian summarization systems available.

Finally, Fig. 5 shows the average rankings by human reviewers of the summaries created by 5 summarizers (2 by humans, 3 by systems).

It can be seen that in most of the diagrams and especially on the average one, Parsumist displays better results than those of the other two summarizers and are

close to human summaries and even in some cases have been ranked better than human summaries by the reviewers.

CONCLUSION

This paper presents single-document-and multi-document summarization methods using lexical chains and graphs. In this system, we use statistical methods combined with some heuristic methods to extract important sentences from the inputs. It also exploits conceptual relations to rank sentences and compute the similarities. According to our experiments, our system yields the best performance among the Persian summarization systems, is the closest to human summaries. Eliminating the shortcomings of Persian processing tools and constructing a Persian lexical ontology can improve the performance of the system. Bestowing some consideration on handling of input files such as newspaper articles or stories can improve the score of the proposed method further. Eliminating redundancies within a sentence, eliminating the gaps in the summary, enhancing the coherence of text and moving toward automatic evaluation of summarization are some of the tasks we have set for ourselves for our further works to improve the system.

REFERENCES

1. Hovy, E. and C. Lin, 1999. Automated Text Summarization and the SUMMARIST System. In Advances in Automatic Text Summarization. Mani, I. and M.T. Maybury (Eds.), MIT Press, pp: 81-94.
2. Edmundson, H.P., 1969. New methods in automatic extracting. Journal of the ACM, 16 (2): 264-285.
3. Jing, H. and K. McKeown, 2000. Cut and paste based text summarization. Proceedings of NAACL'00, Seattle, Washington.
4. Kupiec, J. and J. Pedersen and F. Chen, 1995. A Trainable Document Summarizer. In Proceedings of ACM-SIGIR'95, Seattle, WA.
5. Barzilay, R. and M. Elhadad, 1997. Using Lexical Chains for Text Summarization. In Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain.
6. McKeown, K., S. Chang, J. Cimino, S. Feiner, C. Friedman, I. Gravano, V. Hatzivassiloglou, S. Johnson, D. Jordan, J. Klavans, A. Kushniruk, V. Patel and S. Teufel, 2001. PERSIVAL: A System for Personalized Search and Summarization over Multimedia Healthcare Information.ACM/IEEE Joint Conference On Digital Libraries, Roanoke,VA, pp: 331-340.
7. Elhadad, N., M.Y. Kan, J. Klavans and K. McKeown, 2005. Customization in a unified framework for summarizing medical literature. Artificial Intelligence in Medicine, 33 (2): 179-198.
8. Mani, I., 2001. Automatic Summarization. John Benjamins Pub. Co.
9. Radev, D. and H. Jing and D.T. Malgorzata Stys, 2003. Centroid-Based Summarization of Multiple Documents. Information Processing and Management.
10. Goldstein, J., V. Mittal, J. Carbonell and M. Kantrowitzt, 2000. Multi-document summarization by sentence extraction. In Proc. of the ANLP/NAACL Workshop on Automatic Summarization.
11. Mazdak, N., 2004. FarsiSum-a persian text summarizer. Master thesis,Department of linguistics, Stockholm University.
12. Dalianis, H., 2000. SweSum-A Text Summarizer for Swedish, Technical report", TRITANA-P0015, IPLab-174.
13. Karimi, Z. and M. Shamsfard, 2006. Summarization of Persian texts. In Proceedings of 11th International CSI computer Conference, Tehran, Iran.
14. Honarpisheh, M.A., G. Ghasem-sani and G. Mirroshandel, 2008. A Multi-Document Multi-Lingual Automatic Summarization System. Proceedings of the 3rd Joint Conference on Natural Language Processing, pp: 733-738.
15. Kiani, S. and M. Shamsfard, 2009. Determining the Boundaries of Words and Phrases in Persian written Texts. 14th CSI Computer Conference, Tehran, Iran.