A Novel Method of Significant Words Identification in Text Summarization

Maryam Kiabod
Department of Computer Engineering, Najafabad Branch, Islamic azad University, Isfahan, Iran
Email: m_kiabod@sco.iaun.ac.ir

Mohammad Naderi Dehkordi and Sayed Mehran Sharafi
Department of Computer Engineering, Najafabad Branch, Islamic azad University, Isfahan, Iran
Email: naderi@iaun.ac.ir, mehran_sharafi@iaun.ac.ir

Abstract—**Text summarization is a process that reduces the size of the text document and extracts significant sentences from a text document. We present a novel technique for text summarization. The originality of technique lies on exploiting local and global properties of words and identifying significant words. The local property of word can be considered as the sum of normalized term frequency multiplied by its weight and normalized number of sentences containing that word multiplied by its weight. If local score of a word is less than local score threshold, we remove that word. Global property can be thought of as maximum semantic similarity between a word and title words. Also we introduce an iterative algorithm to identify significant words. This algorithm converges to the fixed number of significant words after some iterations and the number of iterations strongly depends on the text document. We used a two-layered backpropagation neural network with three neurons in the hidden layer to calculate weights. The results show that this technique has better performance than MS-word 2007, baseline and Gistsumm summarizers.**

Index Terms—**Significant Words, Text Summarization, Pruning Algorithm**

I. Introduction

As the amount of information grows rapidly, text summarization is getting more important. Text summarization is a tool to save time and to decide about reading a document or not. It is a very complicated task. It should manipulate a huge quantity of words and produce a cohesive summary. The main goal in text summarization is extracting the most important concept of text document. Two kinds of text summarization are: Extractive and Abstractive. Extractive method selects a subset of sentences that contain the main concept of text. In contrast, abstractive method derives main concept of text and builds the summarization based on Natural Language Processing. Our technique is based on extractive method. There are several techniques used for extractive method. Some researchers applied statistical criterions. Some of these criterions include TF/IDF (Term Frequency-Inverse Document Frequency) [1], number of words occurring in title [2], and number of numerical data [3]. Using these criterions does not produce a reader-friendly summary. As a result NLP (Natural Language Processing) and lexical cohesion [4] are used to guarantee the cohesion of the summary. Lexical cohesion is the chains of related words in text that capture a part of the cohesive structure of the text. Semantic relations between words are used in lexical cohesion. Halliday and Hasan [5] classified lexical cohesion into two categories: reiteration category and collocation category. Reiteration category considers repetition, synonym, and hyponyms, while collocation category deals with the co-occurrence between words in text document. In this article, we present a new technique which benefits of the advantages of both statistical and NLP techniques and reduces the number of words for Natural Language Processing. We use two statistical features: term frequency normalized by number of text words and number of sentences containing the word normalized by total number of text sentences. Also we use synonym, hyponymy, and meronymy relations in reiteration category to reflect the semantic similarity between text words and title words. A two-layered backpropation neural network is used to automate identification of weights of features. The rest of the article is organized as follow. Section 2 provides a review of previous works on text summarization systems. Section 3 presents our technique. Section 4 describes experimental results and evaluation. Finally we conclude and suggest future work in section 5.

II. Text Summarization Approaches

Automatic text summarization dates back to fifties. In 1958, Luhn [6] created text summarization system based on weighting sentences of a text. He used word frequency to specify topic of the text document. There are some methods that consider statistical criterions. Edmundson [7] used Cue method (i.e. "introduction", "conclusion", and "result"), title method and location method for determining the weight of sentences. Statistical methods suffer from not considering the cohesion of text.

Kupiec, Pederson, and Chen [8] suggested a trainable method to summarize text document. In this method,

doi:10.4304/jetwi.4.3.252-258

number of votes collected by the sentence determines the probability of being included the sentence in the summary.

Another method includes graph approach proposed by Kruengkrai and Jaruskululchi [9] to determine text title and produce summary. Their approach takes advantages of both the local and global properties of sentences. They used clusters of significant words within each sentence to calculate the local property of sentence and relations of all sentences in document to determine global property of text document.

Beside statistical methods, there are other approaches that consider semantic relations between words. These methods need linguistic knowledge. Chen, Wang, and Guan [10] proposed an automated text summarization system based on lexical chain. Lexical chain is a series of interrelated words in a text. WordNet is a lexical database which includes relations between words such as synonym, hyponymy, meronymy, and some other relations.

Svore, Vander Wende and Bures [11] used machine learning algorithm to summarize text. Eslami, Khosravyan D., Kyoomarsi, and Khosravi proposed an approach based on Fuzzy Logic [12]. Fuzzy Logic does not guarantee the cohesion of the summary of text. Halavati, Qazvinian, Sharif H. applied Genetic algorithm in text summarization system [13]. Latent Semantic Analysis [14] is another approach used in text summarization system. Abdel Fattha and Ren [15] proposed a technique based on Regression to estimate text features weights. In regression model a mathematical function can relate output to input variables. Feature parameters were considered as input variables and training phase identifies corresponding outputs.

There are some methods that combine algorithms, such as, Fuzzy Logic and PSO [16]. Salim, Salem Binwahla, and Suanmali [17] proposed a technique based on fuzzy logic. Text features (such as similarity to title, sentence length, and similarity to keywords, etc.) were given to fuzzy system as input parameters.

Ref. [18] presented MMR (Maximal Marginal Relevance) as text summarization technique. In this approach a greedy algorithm is used to select the most relevant sentences of text to user query. Another aim in this approach is minimizing redundancy with sentences already included in the summary. Then, a linear combination of these two criterions is used to choose the best sentences for summary. Carbonell and Goldstein [19] used cosine similarity to calculate these two properties. In 2008 [20] used centroid score to calculate the first property and cosine similarity to compute the second property. Different measures of novelty were used to adopt this technique [21, 22]. To avoid greedy algorithms problems, many have used optimization algorithms to solve the new formulation of the summarization task [23, 24, 25].

III. Proposed Technique

The goal in extractive text summarization is selecting the most relevant sentences of the text. One of the most important phases in text summarization process is identifying significant words of the text. Significant words play an important role in specifying the best sentences for summary. There are some methods to identify significant words of the text. Some methods use statistical techniques and some other methods apply semantic relations between words of the text to determine significant words of text. Such as term frequency (TF), similarity to title words, etc. each method has its own advantages and disadvantages. In our work, a combination of these methods is used to improve the performance of the text summarization system. In this way, we use the advantages of several techniques to make text summarization system better. We use both statistical criterions and semantic relations between words to identify significant words of text. Our technique has five steps: preprocessing, calculating words score, significant words identification, calculating sentences score, and sentence selection. These steps are shown in Fig. 1.

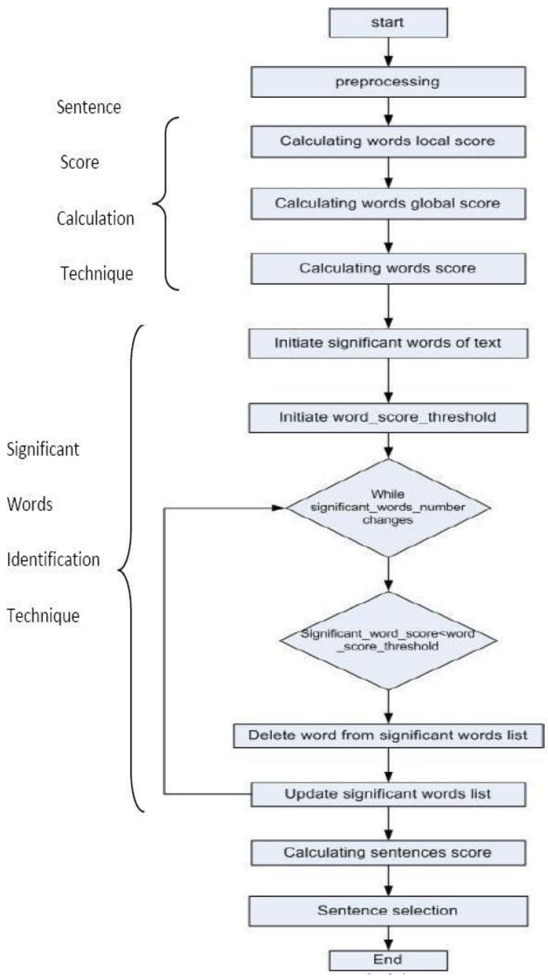

Figure 1: the flowchart of proposed technique

The first step, preprocessing, involves preparing text document for the next analysis and pruning the words of the text document. This step involves sentence segmentation, sentence tokenization part of speech tagging, and finding the nouns of the text document. Keywords or significant words are usually nouns, so finding nouns of the text can help improving performance of our system. The second step, calculating words scores, calculates words scores according to their local score and global score explained in detail later. Local score is determined based on statistical criterions and global score is determined through semantic similarity between a word and title words. The third step, significant words identification, uses words score and an iterative algorithm to select the most important words of text. The fourth step, calculating sentence score, calculates sentence score according to sentence local score, sentence global score and sentence location. The fifth step, sentence selection, selects the most relevant sentences of text based on their scores. These five steps are explained in detail in the next five sections.

A. Preprocessing

The first step in text summarization involves preparing text document to be analyzed by text summarization algorithm. First of all we perform sentence segmentation to separate text document into sentences. Then sentence tokenization is applied to separate the input text into individual words. Some words in text document do not play any role in selecting relevant sentences of text for summary, Such as stop words ("a", "an", the"). For this purpose, we use part of speech tagging to recognize types of the text words. Finally, we separate nouns of the text document. Our technique works on nouns of text. In the rest of the article we use "word" rather than "noun".

B. Calculating Words Score

After preparing input text for text summarization process, it is time to determine words score to be used in later steps. In this step we utilize combination of statistical criterions and lexical cohesion to calculate text words scores. Finding semantic relations between words is a complicated and time consuming process. So, first of all, we remove unimportant words. For this reason, we calculate local score of word. If local score of a word is less than the word_local_score_threshold, we will remove that word. Word_local_score_threshold is the average of all text words scores multiplied by a PF (a number in the range of (0, 1) as a Pruning Factor in word selection). By increasing PF, more words will be removed from text document. In this way, the number of words decreases and the algorithm gets faster. We calculate global score for remaining words based on reiteration category of lexical cohesion. Finally, we calculate words scores by using local and global score of words. This step is described in detail in three next sections.

- Calculating local score of words

In this phase, we use two statistical criterions: term frequency of the word normalized by total number of words (represented by TF) and number of sentences containing the word normalized by total number of sentences of text document (represented by *Sen_Count)*. We combine these two criterions to define equation (1) to calculate local score of words.

$$\text{word\_local\_score} = \alpha * \text{TF} + (1-\alpha) * \text{Sen\_Count} \quad (1)$$

where α is weight of the parameter and is in the range of (0, 1).

We utilize a two-layered backpropagation neural network with three neurons in hidden layer, maximum error of 0.001, and learning rate of 0.2 to obtain this weight. The dendrites weights of this network are initialized in the range of (0, 1). We use sigmoid function as transfer function. The importance of each parameter is determined by the average of dendrites weights connected to the input neuron that represents a parameter [26]. After training neural network with training dataset we use weights to calculate words local scores. The algorithm in this step prunes words of the text document and deletes words without any role in selecting relevant sentences for summary. This is done by defining a threshold and taking words whose scores are above that threshold. This algorithm is shown in Algorithm 1:

Algorithm 1: Word pruning Algorithm

Input*: local score of words, words list*
Output*: pruned words list*

1. $word\_local\_score\_threshold := \frac{\sum_i word\_local\_score(i)}{number\ of\ text\ words} * PF$
2. **foreach** *words w of text* **do**
3. **If** *(word_local_score < word_local_score_threshold)*
   Delete word from significant words list;
4. **end**
5. **end**
6. **return** *pruned words list;*

In this algorithm, *i* represents word index and *PF* stands for Pruning Factor.

The first line of the Algorithm 1 computes local score threshold of words by taking the average of the local score of words multiplied by PF. The second line of it prunes words by taking words whose scores are above the word_local_score_threshold. Finally, the algorithm returns the pruned words list in the seventh line.

- Calculating global score of words

In this phase, we consider semantic similarity between text words and title words. We use WordNet, a lexical database, to determine semantic relations between text words and title words. We fixed the weight of repetition and synonym to 1, of hyponymy and hyperonymy to 0.7, and of meronymy and holonymy to 0.4. We also consider repetition of keywords in the text and fix the weight of it to 0.9.

We define equation (2) to calculate global score of words:

$$\text{Word\_global\_score} = \text{Max}(\text{sim}(w, t_i)) \qquad (2)$$

According to this equation, first of all, we calculate the maximum similarity between each word and title words. Then the sum of maximum similarities is calculated to determine global score of words. This score is used in the next section.

- Calculating word score

The final phase in this step is calculating word score. In our technique, word score is calculated by combination of local score and global score of word. We define equation (3) to calculate word score.

$$\text{Word\_score} = \alpha * (\text{word\_local\_score}) + \beta * (\text{word\_global\_score}) \qquad (3)$$

α and β are determined by neural network illustrated before.

C. Identifying Significant Words

Significant words play an important role in text summarization systems. The sentences containing important words have better chance to be included in summary. In the case of finding significant words of text with a high accuracy, the results of text summarization will be great. So, we focus on significant word identification process to improve text summarization results. In this step, we introduce a new iterative method to determine significant words of text. In this method, significant words are initiated with text words. Then a threshold is defined to be used to identify the words that should be removed from initial significant words. This is done by applying the average of all significant words scores in previous iteration as word_score_threshold. If a word score is less than this threshold, we will remove that word from significant words list. In each loop of this algorithm some words are deleted from significant words list. The algorithm converges to the fixed number of significant words after some iteration. The algorithm is shown below:

Algorithm 2: Significant words identification algorithm

Input: *text words list, text words scores*
Output: *significant words list*

1. *significant_words := text_words;*
2. *Word_score_threshold :=average(text_words_scores);*
3. **while** *number of significant words changes* **do**
4.   **foreach** *significant words of text* **do**
5.   **if** *(word_score< words_score_threshold)*
6.   *Delete word from significant words list;*
7.   **end**
8.   **end**
9.   *Word_score_threshold:=average(significant_words_scores);*
10. **end**
11. **return** *significant words list;*

words_score_threshold in Algorithm 2 is the average of all scores of significant words of text. This threshold changes in every iteration of algorithm. The new value of it is calculated through the average of scores of significant words in previous iteration of algorithm.

The first line of Algorithm 2 initiates significant words list by text words. The second line initiates Word_score_threshold by calculating the average of scores of text words. The third line to the tenth line iterates to delete unimportant words from significant words list. The ninth line of the algorithm computes words_score_threshold for the next iteration. Finally, the algorithm returns significant words list in line ten.

D. Calculating Sentence Score

In this step, we use significant words determined by previous step to calculate sentence score. Our technique in this phase is based on Kruengkrai and Jaruskululchi [9] approach, but we changed the parameters. They combined local and global properties of a sentence to determine sentence score as follow:

$$\text{Sentence\_score} = \alpha * G + (1-\alpha) * L \qquad (4)$$

Where G is the normalized global connectivity score and L is the normalized local clustering score. It results this score in the range of (0, 1).

We define G and L as follow:

$$G = \frac{\sum_i sim(w, t_i)}{total\ number\ of\ words\ in\ sentence}, i = 0, \ldots number\ of\ title_{words} - 1 \qquad (5)$$

$$L = \frac{number\ of\ significant\ words\ of\ sentence}{total\ number\ of\ significant\ words\ of\ text} \qquad (6)$$

where $sim(w, t_i)$ is the maximum semantic relation among sentence words and title and keywords. As shown in equation (5), we consider semantic relations among sentence words and title and keywords to determine the global property of a sentence. Then, we normalize it by total number of words in the sentence. The parameter α determines the importance of G and L. we use neural network illustrated before to determine α.

Baxendale [27] showed that sentences located at first and last paragraph of text document are more important and having greater chances to be included in summary. So, we divide text document into three sections and multiply sentences scores in the first and last section by 0.4 and in the second section by 0.2. The algorithm is shown below.

Algorithm 3: Sentence score calculation algorithm

Input: number of significant words of each sentence, total number of significant words of text, total number of words in each sentence, similarity score between a word and title words, sentence location, and the parameters α and β
Output: scores of sentences

1. **foreach** sentence of text **do**
2. sentence_local_score:= $\frac{\text{number of significant words of sentence}}{\text{total number of significant words of text}}$;
3. sentence_global_score := $\frac{\sum_i \text{sim}(w, t_i)}{\text{total number of words in sentence}}$, $i = 0, \ldots$ number of $\text{title}_{\text{words}} - 1$;
4. Sentence_score := α*G + (1-α)*L;
5. **If** ((1/3)*TSN < sentence_loc < (2/3)*TSN)
6. Sentence_score *:=0.2;
7. **else** : Sentence_score *:=0.4;
8. **end**
9. **end**
10. **return** *scores of sentences;*

TSN IN Algorithm 3 is referred as total number of text sentences. Sentence_loc is the location of sentence in text document.

The Algorithm 3 repeats line two to line eight for each sentence. Line two computes local score of sentences. The third line of the algorithm computes global score of sentence. The forth line computes sentence score according to local score and global score. The fifth line to the eighth line considers the sentence location. If sentence location is in the first section or last section of the text document, multiply it's score by 0.4 otherwise multiply score of sentence by 0.2. Finally, the algorithm returns sentences scores in line ten.

E. Sentence Selection

After calculating scores of the sentences, we can use these scores to select the most important sentences of text. This is done by ranking sentences according to their scores in decreasing order. Sentences with higher score tend to be included in summary more than other sentences of the text document. In our technique these sentences have more similarity to title. This similarity is measured according to statistical and semantic techniques used in our technique. Another criterion to choose sentences for summary is Compression Rate. Compression rate is a scale to decrease the size of text summary. A higher compression rate leads to a shorter summary. We fix compression rate to 80%. Then n top-scoring sentences are selected according to compression rate to form the output summary.

IV. Evaluation

Text summarization evaluation is a complicated task. We use three criterions to evaluate our system [28]:

$$\textit{Precision Rate} = \frac{\textit{number of correctly selected sentences}}{\textit{total number of selected sentences}} \quad (7)$$

$$\textit{Recall Rate} = \frac{\textit{number of correctly selected sentences}}{\textit{total number of correct sentences}} \quad (8)$$

$$\textit{F-measure} = \frac{2 * precision * recall}{precision + recall} \quad (9)$$

We use DUC2002 [1] as input data to train neural network and test our technique. DUC 2002 is a collection of newswire articles, comprised of 59 document clusters. Each document in DUC2002 consists of 9 to 56 sentences with an average of 28 sentences. Each document within the collections has one or two manually created abstracts with approximately 100 words which are specified by a model.

We evaluate the technique for different PF. The best result was achieved for PF=0.25 as shown in Fig. 2. We compare our results with MS-word 2007, Gistsumm, and baseline summarizers. MS-word 2007 uses statistical criterions, such as term frequency, to summarize a text. Gistsumm uses the gist as a guideline to identify and select text segments to include in the final extract. Gist is calculated on the basis of a list of keywords of the source text and is the result of the measurement of the representativeness of intra- and inter-paragraph sentences. The baseline is the first 100 words from the beginning of the document as determine by DUC2002.

The results are shown in Fig. 3 and Fig. 4. The numerical results are shown in Table 1. The text number in Table 1 shows the text number in the tables. Our technique (OptSumm) reaches the average precision of 0.577, recall of 0.4935 and f-measure of 0.531. The MS-word 2007 summarizer achieves the average precision of 0.258, recall of 0.252 and f-measure of 0.254. The Gistsumm reaches the average precision of 0.333 and f-measure of 0.299. the baseline achieves the average of 0.388, recall of 0.28 and f-measure of 0.325.the results have shown that our system has better performance in comparison with MS-word 2007, Gistsumm and baseline summarizers.

Fig. 3, Fig. 4, and Fig. 5 show that the precision score, the Recall score, and F-measure are higher when we use OptSumm rather than MS-word 2007, Gistsumm, and baseline summarizers.

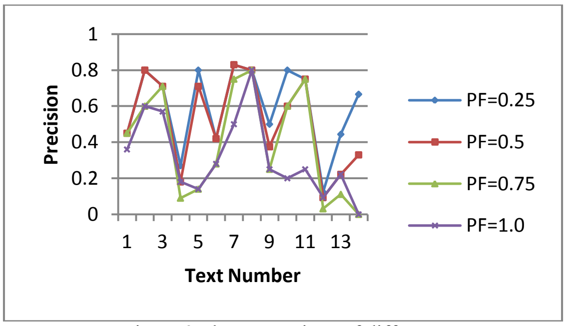

Figure 2: the comparison of different PF

1. www.nlpir.nist.gov

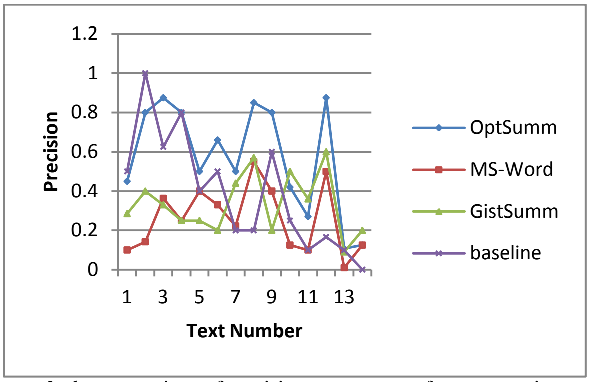

Figure 3: the comparison of precision score among four summarizers

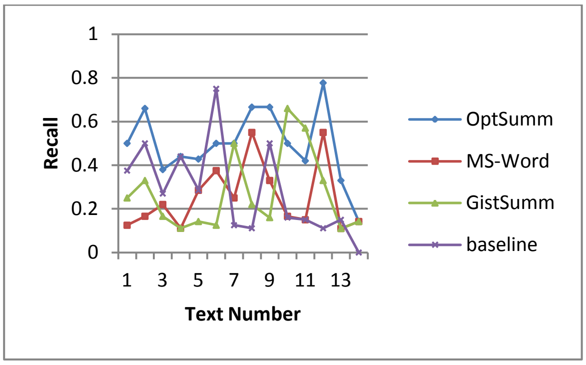

Figure 4: the comparison of recall score among four summarizers

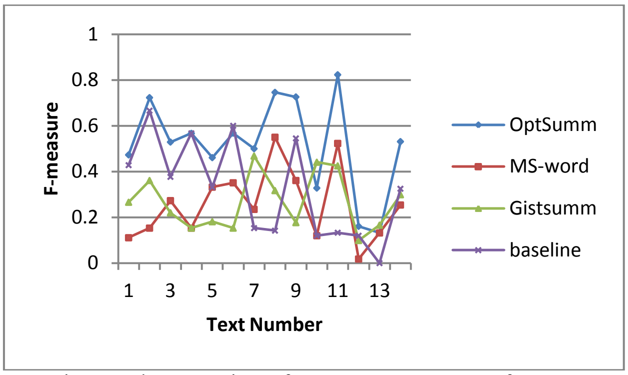

Figure 5: the comparison of F-measure score among four summarizers

Table I.
THE COMPARISON OF PRECISION AND RECALL AMONG FOUR SUMMARIZERS

Text Number	SET NO.	Model	OptSumm			MS-word 2007			GistSumm			baseline		
			Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
1	D061J	b	0.45	0.5	0.473	0.1	0.125	0.111	0.285	0.25	0.266	0.5	0.375	0.428
2	D062J	a	0.8	0.66	0.723	0.142	0.166	0.153	0.4	0.33	0.361	1.0	0.5	0.666
3	D106g	a	0.875	0.38	0.529	0.363	0.22	0.273	0.33	0.166	0.220	0.625	0.27	0.377
4	D113h	b	0.8	0.44	0.567	0.25	0.111	0.153	0.25	0.111	0.153	0.8	0.44	0.567
5	D083a	b	0.5	0.428	0.461	0.4	0.285	0.332	0.25	0.142	0.181	0.4	0.285	0.332
6	D071f	a	0.66	0.5	0.568	0.33	0.375	0.351	0.2	0.125	0.153	0.5	0.75	0.6
7	D072f	j	0.5	0.5	0.5	0.222	0.25	0.235	0.44	0.5	0.468	0.2	0.125	0.153
8	D092c	a	0.85	0.666	0.746	0.55	0.55	0.55	0.57	0.22	0.317	0.2	0.111	0.142
9	D074b	a	0.8	0.666	0.726	0.4	0.33	0.361	0.2	0.16	0.177	0.6	0.5	0.545
10	D091c	j	0.27	0.42	0.328	0.1	0.15	0.12	0.36	0.57	0.441	0.1	0.15	0.12
11	D110h	b	0.875	0.777	0.823	0.5	0.55	0.523	0.6	0.33	0.425	0.166	0.111	0.133
12	D102e	f	0.107	0.33	0.161	0.01	0.11	0.018	0.09	0.11	0.099	0.1	0.15	0.12
13	D098e	a	0.125	0.142	0.132	0.125	0.142	0.132	0.2	0.142	0.166	0	0	0
14	average	-	0.577	0.4935	0.531	0.258	0.252	0.254	0.333	0.272	0.299	0.388	0.28	0.325

V. CONCLUSION and FUTURE WORK

In this article, we proposed a new technique to summarize text documents. We introduced a new approach to calculate words scores and identify significant words of the text. A neural network was used to determine the style of human reader and to which words and sentences the human reader deems to be important in a text. The evaluation results show better performance than MS-word 2007, GistSumm, and baseline summarizers. In future work, we intend to use other features, such as font based feature and cue-phrase feature in words local score and calculate words scores based on it. Also the sentence local score and global score can be changed to reflect the reader's needs.

REFERENCES

[1] M.Wasson, "Using Leading Text for News Summaries: Evaluation results and implications for commercial summarization applications", In Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the ACL, pp.1364-1368, 1998.

[2] G.Salton,C.Buckley,"Term-weighting Approaches in Automatic Text Retrieval", Information Proceeding and Management 24,1988,513-523.Reprinted in:Sparck-Jones, K.; Willet ,P.(eds).Readings in I.Retreival, Morgan Kaufmann,pp.323-328,1997

[3] C.Y.Lin, "Training a Selection Function for Extraction", In Proceedings of eighth international conference on Information and knowledge management, Kansas City, Missouri, United States, pp.55-62,1999.

[4] M.Hoey, Patterns of Lexis in Text. Oxford: Oxford University Press, 1991

[5] M.Halliday, and Hasan, R.1975.Cohesion in English. London: Longman
[6] H.P.Luhn, "The Automatic Creation of Literature Abstracts", IBM journal of Research Development, 1958, pp.159-165.
[7] H.P.Edmundson, "New Methods in Automatic Extraction", journal of the ACM, 1969, pp.264-285.
[8] J.Kupiec , j.Pedersen, AND j.Chen, "A Trainable Document Summarizer", In Proceedings of the 18th ACMSIGIR Conference,1955,pp.68-73.
[9] C.Jaruskululchi, Kruengkrai, "Generic Text Summarization Using Local and Global Properties of Sentences", IEEE/WIC international conference on web intelligence, October 2003, pp.13-16.
[10] Y.Chen, X. Wang, L.V.YI Guan," Automatic Text Summarization Based on Lexical Chains", in Advances in Natural Computation, 2005, pp.947-951.
[11] K.Svore, L.Vanderwende, and C.Bures, "Enhancing Single-document Summarization by Combining Ranknet and Third-party Sources", In Proceeding of the EMNLP-CoNLL.
[12] F.Kyoomarsi, H.Khosravi, E.Eslami, and P.Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In Proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, University of shahid Bahonar Kerman,2008,pp.347-352.
[13] V.Qazvinian, L.Sharif Hassanabadi, R.Halavati, "Summarization Text with a Genetic Algorithm-Based Sentence Extraction", International of Knowledge Management Studies (IJKMS),2008,vol.4,no.2,pp.426-444.
[14] S.Hariharan, "Multi Document Summarization by Combinational Approach", International Journal of Computational Cognition, 2010, vol.8, no.4, pp.68-74.
[15] M.Abdel Fattah, and F.Ren, "Automatic Text Summarization", Proceedings of World of Science, Engineering and Technology,2008,vol.27,pp.195-192.
[16] L.Suanmali, M.Salem, N.Binwahlan and Salim, "sentence Features Fusion for Text Summarization Using Fuzzy Logic", IEEE, 2009,pp.142-145.
[17] L.Suanmali, N. Salim, and M.Salem Binwahlan, "Fuzzy Swarm Based Text Summarization", journal of computer science, 2009, pp.338-346.
[18] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based rerunning for reordering documents and producing summaries," in Proceedings of theAnnual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336, 1998.
[19] C. D. Manning and H. Schutze, Foundations of Natural Language Processing.MIT Press, 1999.
[20] S. Xie and Y. Liu, "Using corpus and knowledge-based similarity measure in Maximum Marginal Relevance for meeting summarization," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4985–4988, 2008.
[21] G. Murray, S. Renals, and J. Carletta, "Extractive summarization of meeting recordings," in Proceedings of 9th European Conference on Speech Communication and Technology, pp. 593–596, 2005.
[22] D. R. Timothy, T. Allison, S. Blair-goldensohn, J. Blitzer, A. elebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, A. Winkel, and Z. Zhang, "MEAD — a platform for multidocument multilingual text summarization," in Proceedings of the International Conference on Language Resources and Evaluation, 2004.
[23] R. McDonald, "A study of global inference algorithms in multi-document summarization," in Proceedings of the European Conference on IR Research, pp. 557–564, 2007.
[24] S. Ye, T.-S. Chua, M.-Y. Kan, and L. Qiu, "Document concept lattice for text understanding and summarization," Information Processing and Management, vol. 43, no. 6, pp. 1643–1662, 2007.
[25] W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki, "Multi-document summarization by maximizing informative content-words," in Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1776–1782, 2007.
[26] N.Soltanian Zadeh, L.Sharif, "Evaluation of Effective Parameters and Their Effect on Summarization Systems Using Neural Network", Fifth annual international conference of computer society of iran, 2008.
[27] P.Baxendale," Machine-Made Index for Technical Literature –an Experiment", IBM Journal of Research Development,1958.
[28] Y.Y. Chen, O.M.Foong, S.P.Uong, I.Kurniawan, "Text Summarization for Oil and Gas Drilling Topic", Proceeding of world academy of science and technology, vol.32, pp.37-40, 2008.