GENETIC ALGORITHM BASED SENTENCE EXTRACTION FOR TEXT SUMMARIZATION

Ladda Suanmali[1], Naomie Salim[2], Mohammed Salem Binwahlan[3]

[1]Faculty of Science and Technology, Suan Dusit Rajabhat University

Dusit, Bangkok, 10300 Thailand

[2]Faculty of Computer Science and Information Systems,

Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[3]Faculty of Applied Sciences, Hadhramout University of Science and Technology

Seiyun, Hadhramout, Yemen

email: [1]nongnu_@hotmail.com, [2]naomie@utm.my, [3]moham2007med@yahoo.com

ABSTRACT

The goal of text summarization is to generate summary of the original text that helps the user to quickly understand large volumes of information available in that text. This paper focuses on text summarization based on sentence extraction. One of the methods to obtain suitable sentences is to assign some numerical measure for sentences called sentence weighting and then select the best ones. The first step in summarization by extraction is the identification of important features. In this paper, we consider the effectiveness of the features selected using Genetic Algorithm (GA). GA is used for the training of 100 documents in DUC 2002 data set to learn the weight of each feature, which is evaluated using recall measurement generated by ROUGE for a fitness function. The weights obtained by GA were used to adjust the important features score. We compare our results with Microsoft Word 2007 summarizer and Copernic summarizer both for 100 documents and 62 unseen documents. The results show that the best average precision, recall, and f-measure for the summaries were obtained by GA.

Key Word: Genetic Algorithm, Sentence extraction, Statistic method, Text summarization

.

1.0 INTRODUCTION

Text summarization has become an important and timely tool for interpreting large volumes of text available in documents. One of the natural questions to ask in doing summarization is “what are the properties of text that should be represented or kept in a summary?”

Text summarization addresses both the problem of finding the most important subset of sentences in text, which in some way represents its source text and the problem of generating coherent summaries. This process is significantly different from human based text summarization since human can capture and relate deep meanings and themes of text documents while automation of such a skill is very difficult to implement. The goal of text summarization is to present the most important information in a shorter version of the original text while keeping its main content and helps the user to quickly understand the large volume of information. Automatic text summarization researchers since Luhn (1958) are trying to solve or at least relieve that problem by proposing techniques for generating summaries. A number of researchers have proposed techniques for automatic text summarization which can be classified into two categories: extraction and abstraction. Extraction summary is produced by selecting sentences or phrases from the original text with the highest score and put it together into a new shorter text without changing the source text. Abstraction summary method uses linguistic methods to examine and interpret the text for generative of abstracts. Most of the current automated text summarization systems use extraction method to produce a summary (Ko and Seo, 2008; Yulia et al., 2008; Suanmali et al., 2009; Ramiz, 2009).
Sentence extraction techniques are commonly used to produce extraction summaries. One of the methods to obtain suitable sentences is to assign some numerical measure of a sentence for the summary called sentence weighting and then select the best sentences to form document summary based on the compression rate. In the extraction method, compression rate is an important factor used to define the ratio between the length of the summary and the source text. As the compression rate increases, the summary will be larger, and more insignificant content is contained. While the compression rate decreases the summary to be short, more information is lost. In fact, when the

compression rate is 5-30%, the quality of summary is acceptable (Fattah and Ren, 2008; Yeh et al., 2005; Mani and Maybury, 1999; Kupiec et al., 1995).

The first step in summarization by extraction is the identification of important features such as sentence length, sentence location (Fattah and Ren, 2008), term frequency (Salton, 1989), number of words occurring in title (Salton and Buckley, 1997), number of proper nouns (Yulia et al., 2008) and number of numerical data (Lin, 1999). In our approach, we utilize a feature fusion technique to discover which features out of the available ones are most useful.

Kiani and Akbarzadeh (2006) proposed technique for summarizing text using a combination of Genetic Algorithm (GA) and Genetic Programming (GP) to optimize rule sets and membership function of fuzzy systems. Vahed et al. (2008) used GA to produce document summary. They proposed a fitness function based on three following factors: Readability Factor (RF), Cohesion Factor (CF) and Topic-Relation Factor (TRF).

In this paper, we propose use genetic algorithm method to extract important sentences as a summary. The rest of this paper is organized as follows. Section 2 describes preprocessing and the important features used. Section 3 and 4 describes our proposed method, followed by experimental design, experimental results and evaluation. Finally, we conclude and suggest future work that can be carried out in Section 5.

2.0 EXPERIMENTAL DESIGN

We used the test document sets (D061j, D062j, D063j, D064j, D065j, D066j, D067f, D068f, D069f, D070f, D071f, D072f, D073b, D074b, D075b, D077b, D078b, D079b, and D080b) from DUC2002 (DUC., 2002) comprising of 162 documents to create automatic document summarization. Each document in DUC2002 collection is supplied with a set of human-generated summaries provided by two different experts. While each expert was asked to generate summaries of different length, we use only generic 100-word variants.

We divide the 162 documents into two groups. The first 100 documents were used for training. The other 62 documents were used to evaluate and compare the results.
Currently, input document is of plain text format. There are four main activities performed in this stage: sentence segmentation, tokenization, stop word removal, and word stemming. Sentence segmentation is performed by boundary detection and separating source text into sentences. Tokenization is done by separating the input document into individual words. Next, words which rarely contribute to useful information in terms of document relevance and appear frequently in document but provide less meaning in identifying the important content of the document are removed. These words include articles, prepositions, conjunctions and other high-frequency words such as *'a', 'an', 'the', 'in', 'and', 'I', etc*... The last step for preprocessing is word stemming. Word stemming is the process of reducing inflected or derived words to their stem, base or root form. In this research, we performed words stemming using Porter's stemming algorithm (Porter, 1980). For example, a stemming algorithm for English should stem from the words *'compute', 'computed', 'computer', 'computable', and 'computation'* to its word stem, *'comput'*.

2.1 SENTENCE FEATURES

After preprocessing, each sentence of the document is represented by an attribute vector of features. These features are attributes that attempt to represent the sentence in the task of sentence selection. We focus on eight features for each sentence. Each feature is given a value between '0' and '1'. There are eight features as follows:

2.1.1 Title feature

The word in sentence that also occurs in the title is given higher score. This is determined by counting the number of matches between the content words in a sentence and the words in the title. We calculate the score for this feature which is the ratio of the number of words in the sentence that occur in the title over the number of words in title.

$$S\_F1(S) = \frac{No.Title\ word\ in\ S}{No.Word\ in\ Title} \quad (1)$$

2.1.2 Sentence Length

This feature is useful to filter out short sentences such as datelines and author names commonly found in news articles. The short sentences are not expected to belong to the summary. We use normalized length of the sentence, which is the ratio of the number of words occurring in the sentence over the number of words occurring in the longest sentence of the document.

$$S\_F2(S) = \frac{No.Word\ occurring\ in\ S}{No.Word\ occurring\ in\ longest\ sentence} \quad (2)$$

2.1.3 Term Weight

The frequency of term occurrences within a document has often been used for calculating the importance of sentence. The score of a sentence can be calculated as the sum of the score of words in the sentence. The score *wi* of word *i* can be calculated by the classic *tf.idf* method as follows (Salton and Buckley, 1997). We applied this method to *tf.isf* (Term frequency, Inverse sentence frequency).

$$w_i = tf_i x isf_i = tf_i x log \frac{N}{n_i} \quad (3)$$

where tf_i is the tern frequency of word *i* in the document, *N* is the total number of sentences, and n_i is number of sentences in which word *i* occurs. This feature can be calculated as follows.

$$S\_F2(S) = \frac{\sum_{i=1}^{k} W_i(S)}{Max\left(\sum_{i=1}^{k} W_i(S_i^N)\right)} \quad (4)$$

where *k* is number of words in sentence.

2.1.4 Sentence Position

A sentence position in the text can indicate importance of the sentence. This feature can involve several items such as the position of a sentence in the document, section, and paragraph, etc.. The first sentence has the highest ranking. We only consider up to 5 positions from the top of the document. For instance, the first sentence in a paragraph has a score value of 5/5, the second sentence has a score 4/5, and so on.

$$S\_F4(S) = 5/5 \text{ for } 1^{st},\ 4/5 \text{ for } 2^{nd},\ 3/5 \text{ for } 3^{rd},$$
$$2/5 \text{ for } 4^{th},\ 1/5 \text{ for } 5^{th},$$
$$0/5 \text{ for other sentences} \quad (5)$$

2.1.5 Sentence to Sentence Similarity

Similarity between sentences, for each sentence *s*, is the similarity between *s* and all other sentences, as computed by the cosine similarity measure. The score of this feature for a sentence *s* is obtained by computing the ratio of the summary of sentence similarity of sentence *s* with all other sentences over the maximum of sentence similarity.

$$S\_F5(S) = \frac{\text{Sum of Sentemce Similarity in S}}{\text{Max(Sum of Sentence Similarity)}} \quad (6)$$

2.1.6 Proper Noun

Usually the sentence that contains more proper nouns is an important one and it is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of proper nouns in the sentence over the sentence length.

$$S\_F6(S) = \frac{\text{No. Proper nouns in S}}{\text{Length (S)}} \quad (7)$$

2.1.7 Thematic Word

The number of thematic word in a sentence is important because terms that occur frequently in a document are probably related to the same topic. The number of thematic words indicates the words with maximum possible relativity. We used the top

10 most frequent content word for consideration as thematic. The score for this feature is calculated as the ratio of the number of thematic words in the sentence over the maximum summary of thematic words in the sentence.

$$S\_F7(S) = \frac{No.\ Thematic\ word\ in\ S}{Max(No.\ Thematic\ word)} \quad (8)$$

2.1.8 Numerical Data

A sentence that contains numerical data is considered important and is most probably included in the document summary. The score for this feature is calculated as the ratio of the number of numerical data in sentence over the sentence length.

$$S\_F8(S) = \frac{No.\ Numerical\ data\ in\ S}{Length\ (S)} \quad (9)$$

2.2 THE METHODS

The features score of each sentence can be calculated as described in the previous section. In this section, we use two methods to extract important sentences: text summarization based on general statistic method (GSM) and Genetic Algorithm (GA).

2.2.1 Text Summarization based on General Statistic Method (GSM)

The feature score of each sentence described in the previous section are used to obtain the significant sentences. In this section, we used general statistic method (GSM) to extract the important sentences. The technique consists of the following main steps.

(1) Read the source document into the system.

(2) For preprocessing, the system extracts the individual sentences of the original documents. Then, the input document is separated into individual words. Next, remove the stop words. The last step of preprocessing is word stemming.

(3) Each sentence is associated with a vector of eight features described in Section 2.1, whose values are derived from the content of the sentence.

(4) The features are calculated to obtain the sentence score based on general statistic method (GSM) shows in Figure 1;

(5) A set of the highest score sentences are extracted as a document summary based on the compression rate.

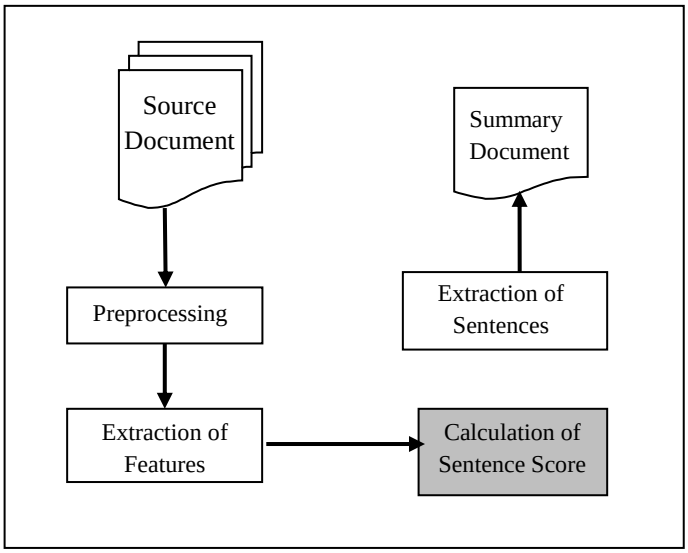

Figure 1: Text summarization based on general statistic method architecture

Text summarization based on general statistical method is produced by using the sentence weight. First, for a sentence *s*, a weighted score function, as shown in the following equation (eq. 10) is exploited to integrate all the eight feature scores mentioned in Section 2.1.

$$Score(S) = \sum_{k=1}^{8} S\_Fk(S) \qquad (10)$$

Where *Score(S)* is the score of the sentence *S* and *S_Fk(S)* is the score of the feature *k*

2.2.1 Text Summarization based on Genetic Algorithm (GA)

Genetic algorithm (GA) provides an alternative to traditional optimization techniques by using directed random searches to locate optimal solutions in complex landscapes. GA generates a sequence of populations by using a selection mechanism, where cross-overs and mutations are used as part of the search mechanisms (Srinivas and Patnaik, 1994), as shown in Figure 2.

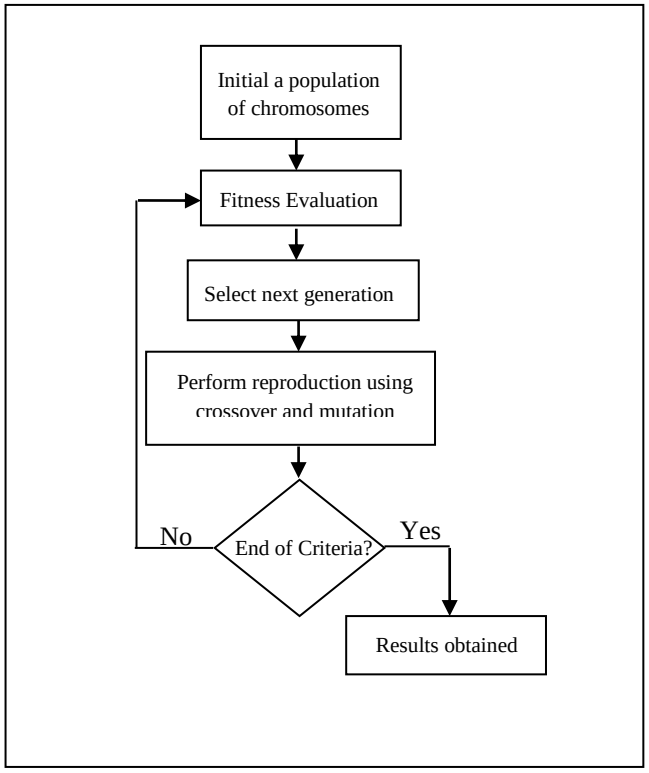

Figure 2: Simple genetic algorithm structure

Chromosome Encoding

Before applying a GA, we first must encode the parameters of the problem to be optimized. GAs work with codes that represent the parameter. There are two common representation methods; floating point and bit string that can be used to represent the parameter. In our research, we use the bit string (containing binary digits: 0s and 1s) because the majority of genetic operators are suitable for our representation. To learn the feature weights, each chromosome contains the genes connected together into a long string represented by a binary vector of dimension F (where F is the total number of features). Each gene represents a specific feature as bit string. Each bit represents a value one or zero for each feature. If the value 1 is represented in the bit, it means that the feature is selected, otherwise the feature is not selected. The first bit refers to the first feature; the second bit refers to the second feature and so on. Each chromosome is represented by a binary vector of dimension F (where F is the total number of features), as shown in the following Figure 3.

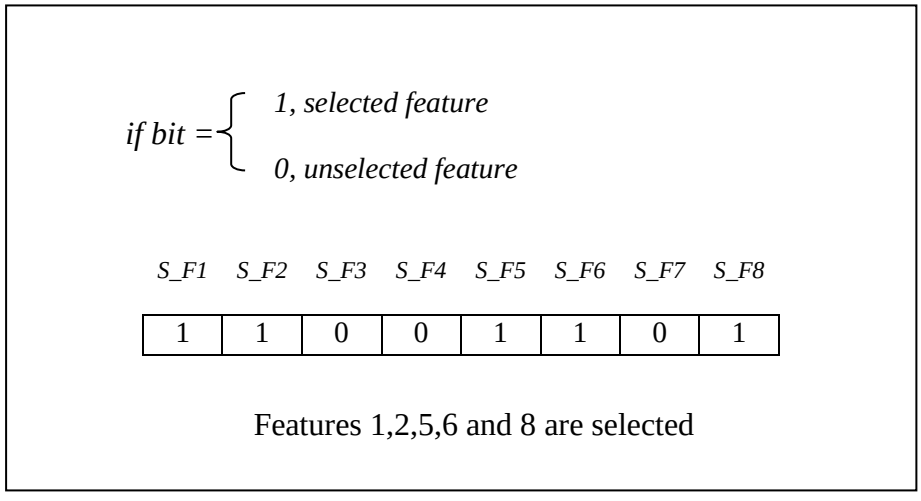

Figure 3: Structure of chromosome

Genetic Algorithm Structure

1. Initial population

Each individual is made up of a sequence of bits (0 and 1). Let *N* be the size of a chromosome population. The population of 50 chromosomes is randomly generated in the beginning. We used random function to generate a random floating-point array [0, 1]. We then used the round function to convert it into an integer, with a bias of 0.4 for 0 (random < 0.5) and 0.6 for 1 (random>=0.5).

2. Fitness function

The fitness value reflects how good a chromosome is compared to the other chromosomes in the population. The higher chromosome has a higher chance of survival and reproduction that can represent the next generation. In this paper, fitness(x) is equal to the average recall from 70 training documents generated by ROUGE (Lin, 2004).

3. Selection

In this state, the existing population is selected to be a new generation through a fitness based process. In each cycle, we chose two parents that give the highest average recall.

4. Crossover and mutation

Crossover

The function of the crossover is to generate new or child chromosomes from two parent chromosomes by combining the information extracted from the parents. In each generation, we generate new chromosomes using two crossover operations: one point crossover and two point crossover. The one point crossover is chosen using a random function. Then we swap the bit strings between two parents from the beginning until the random point. One the other hand, in the two point of crossover, two random points

in the bit string is swapped with the parents from the first random point until the second random point. The example is illustrated in Figure 4.

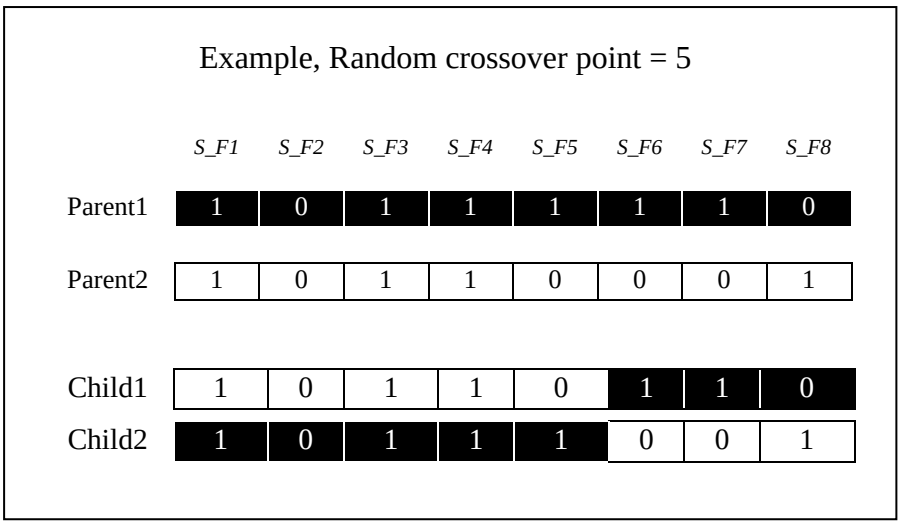

Figure 4: Generated next generation using crossover

Mutation

A mutation operator involves a probability that an arbitrary bit in a chromosome sequence will be changed from an original state. In this paper, we generated new or child chromosomes from two parent chromosomes by changing some parts of the chromosome using bit swapping. We also used single mutation with 1, 2, 3, and 4 digit(s) for the bit swapping using random function. In Figure 5, an example for 2 digits swapping, the two numbers were generated by random function. The random variable tells us whether a particular bit will be modified. In Figure 5, the two random numbers for the selected bits are 2 and 8, which means that the 2^{nd} and 8^{th} bits are swapped from 0 to 1 or 1 to 0.

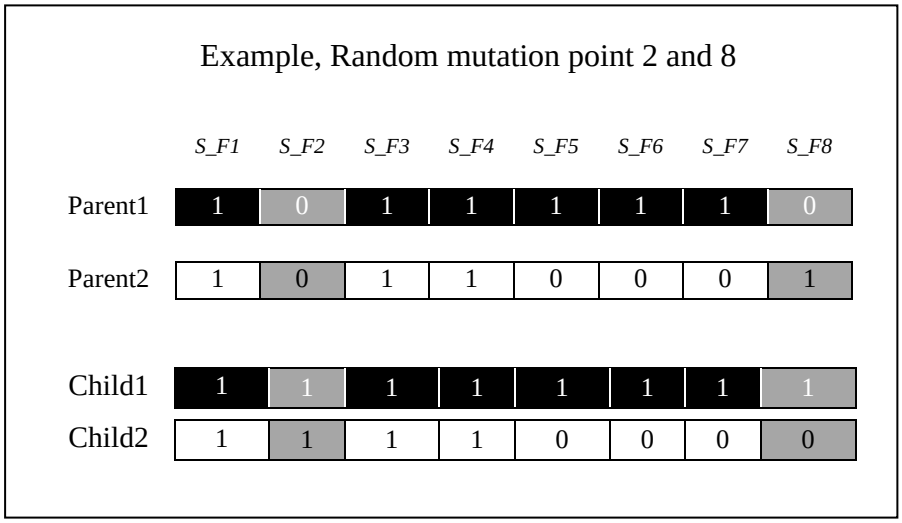

Figure 5: Generated next generation using mutation

These processes will continue until the fitness value of individuals in the population converges.

5. Fitness Function Algorithm

The document sentences are scored using (eq.10) and ranked in a descending order according to their scores. A set of the highest scoring sentences are extracted as a document summary based on the compression rate. In this study, we used a 20% compression rate as summary length. Then, we used an average recall generated by ROUGE-1 (Lin, 2004) as the fitness function, as shown in equation 11

$$\frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1$$

Where n is the length of the n-gram and $count_{match}$ is the highest number of n-grams shared between a systems generated summary and a set of reference summaries.

Calculating Sentence Score

The individual chromosome selected is used to calculate score for each sentence in the documents. The scores of each sentence feature are presented as a vector. The vectors of the sentence features are used as inputs. Finally the sentence score is obtained using this following equation (eq. 12)

$$Score(S) = \sum_{k=1}^{8} Ck * S\_Fk(S) \qquad (12)$$

Where *Score(S)* is the score of the sentence *S* , *S_Fk(S)* is the score of the feature *k* and *Ck* is binary value of feature *k.*

We generate 100 generations and keep the highest fitness value from each generation and then compare all highest fitness values. The best fitness value shows the features that are suitable for a data set. The weights of the document features are calculated as an average of the vectors created in each run. The final features weights are calculated over the vectors of the features weights for all documents in the data collection.

After 100 documents were trained and tested by GA, the average weight of each feature was obtained. We used the average weights to adjust the feature score for 62 unseen documents. The sentence score for new documents can be calculated using the following equation (13).

$$Score(S) = \sum_{k=1}^{8} W_k * S\_Fk(S) \qquad (13)$$

Where *Score(S)* is the score of the sentence *S*, W_i is the average weight of the feature *k* generated by GA and *S_Fk(S)* is the score of the feature *k.*

2.3 EXTRACTION OF SENTENCES

In those two methods, each sentence of the document is represented by a sentence score. All document sentences are then ranked in descending order according to their scores. A set of the highest scoring sentences are extracted as document summary based on the compression rate. Therefore, we extracted the appropriate number of sentences according to 20% compression rate. It has been proven that the extraction of 20% of sentences from the source document can be considered as informative as the full text of a document (Morris et al, 1992). Finally, the summary sentences are arranged in the original order.

3.0 RESULTS

After the 100 documents were trained, we used 62 unseen documents to evaluate the results using the ROUGE, a set of metrics called Recall-Oriented Understudy for Gisting Evaluation. The evaluation toolkit (Lin, 2004) that has become a standard for automatic evaluation of summaries. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses n-gram statistics. Our evaluation was done using n-gram setting of ROUGE, which was found to have the highest correlation with human judgments at a confidence level of 95%. It is claimed that ROUGE-1 consistently correlates highly with human assessments and has a high recall and precision significance test with manual evaluation results. We choose ROUGE-1 as the measurement for our experimental results. In Table 1, we compare the average precision, recall and f-measure score between general

statistic method (GSM), GA method, Microsoft Word 2007 Summarizer, and Copernic summarizer.

Table 1: The comparison of average precision, recall and f-measure score among four summarizers

Summarizer	Training and Testing Data sets (100 Documents)			62 Unseen Documents		
	Avg.P	**Avg.R**	**Avg.F**	**Avg.P**	**Avg.R**	**Avg.F**
GSM	0.49583	0.44216	0.46259	0.46646	0.44318	0.45170
GA Method	**0.49800**	**0.44649**	**0.46622**	**0.46471**	**0.44673**	**0.45359**
MS-Word	0.48189	0.39138	0.42279	0.46967	0.42265	0.43903
Copernic	0.51253	0.40984	0.44647	0.47131	0.42168	0.43975

We compare the results for both 100 documents of training-testing data sets and 62 unseen documents. In the 100 documents, the GSM reaches the average precision of 0.49583, recall of 0.44216 and f-measure of 0.46259. The GA method summarizer achieves the average precision of 0.49800, recall of 0.44649 and f-measure of 0.46622. While Microsoft Word 2007 summarizer reaches the average precision 0.48189, recall of 0.39138 and f-measure of 0.42279 and the Copernic summarizer reaches an average precision of 0.51253, recall of 0.40984 and f-measure of 0.44647.

We also compare the results of 62 unseen documents. The GSM gives the average precision of 0.46646, recall of 0.44318 and f-measure of 0.45170. The GA method summarizer achieves the average precision of 0.46471, recall of 0.44673 and f-measure of 0.45359. While Microsoft Word 2007 summarizer reaches the average precision 0.46967, recall of 0.42265 and f-measure of 0.43903 and the Copernic summarizer reaches an average precision of 0.47131, recall of 0.42168 and f-measure of 0.43975.

The results of the experiment in Figure 6 and 7 confirm that genetic algorithm has a significant improvement in terms of quality of text summary. It is claimed that the results of ROUGE-1 of all summarizers consistently correlate highly with human

assessments and have a high precision, recall and f-measure significance test with evaluation results.

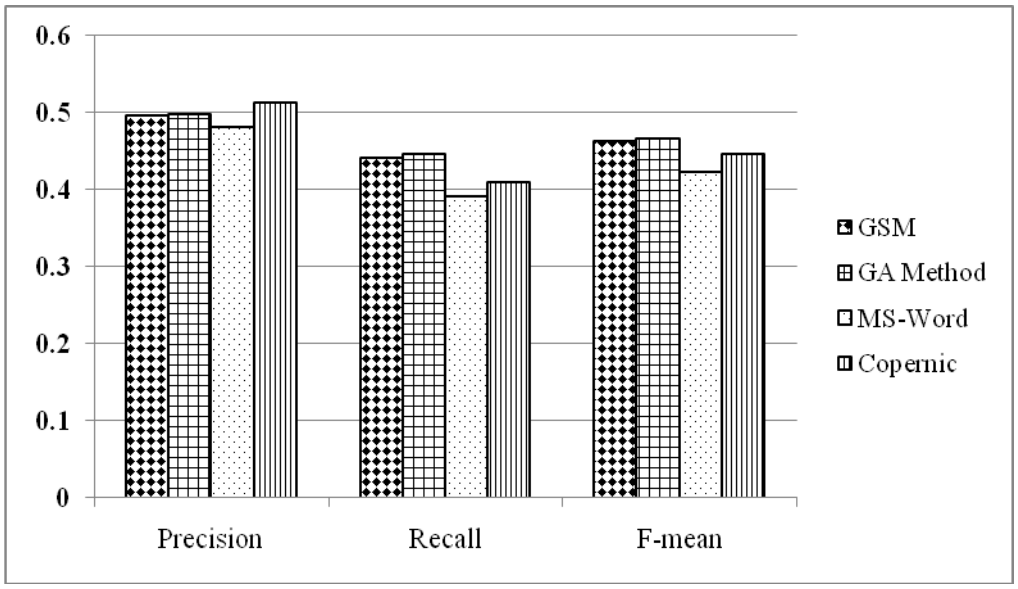

Figure 6: Average precision recall and f-measure score of 100 training documents among four summarizers

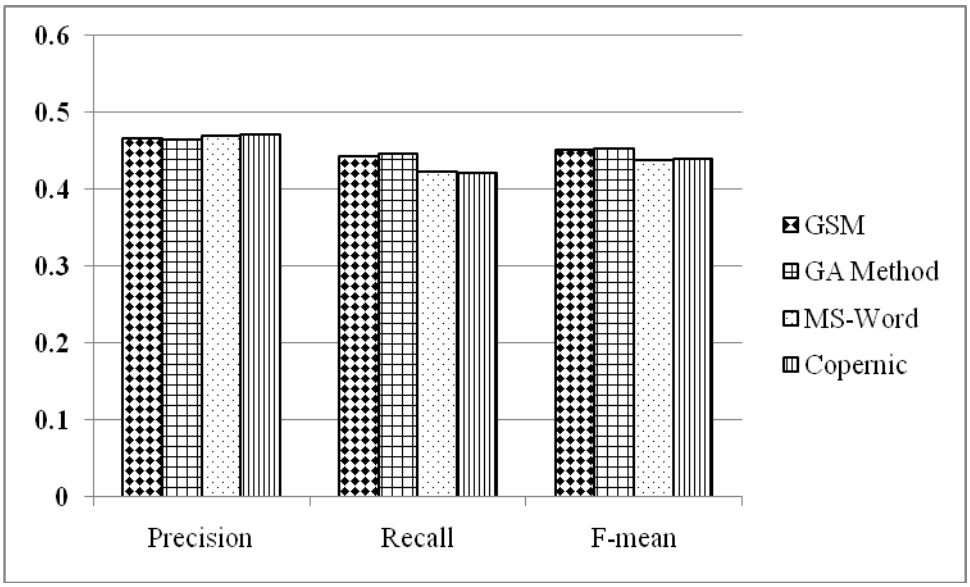

Figure 7: Average precision recall and f-measure score of 62 unseen documents among four summarizers

4.0 DISCUSSION

In this paper, we have presented a method based on general statistic method (GSM) and a genetic algorithm aided sentence extraction summarizer that can be as informative as the full text of a document with good information coverage. A prototype has also been constructed to evaluate this automatic text summarization scheme by using some news articles collection provided by DUC2002 as input. Afterwards, we extracted important features and perform summary document based on those score called GSM. Then, we proposed genetic algorithm (GA) method to adjust each feature. The benefit of GA is to find and optimize the corresponding weight of each feature. Furthermore, GA is used to obtain an appropriate set of feature weights. A chromosome is represented as the combination of all feature weights. In training phrase, we defined fitness as the average recall obtained with the genome. The average feature weights obtained by GA cannot guarantee that the feature weights are best for the test corpus.

5.0 CONCLUSIONS

In this paper, we propose text summarization based on genetic algorithm to improve the quality of summary results based on the general statistic method. We extracted the important features for each sentence of the document and represent them as a vector of features consisting of the following elements: title feature, sentence length, term weight, sentence position, sentence to sentence similarity, proper noun, thematic word and numerical data. We use 162 documents from DUC2002 data set. We divide 162 documents into two groups. The first 100 documents were used for training and testing. The other 62 documents were used to evaluate and compare the results as unseen documents. We compare our summarizer with the Copernic summarizer and Microsoft Word 2007 summarizers. The results show that the genetic algorithm gives significant quality improvement for text summarization.

ACKNOWLEDGEMENTS

This project is sponsored partly by the Ministry of Science, Technology and Innovation under E-Science grant 01-01-06-SF0502, Malaysia. We would like to thank Suan Dusit Rajabhat University and Universiti Teknologi Malaysia for supporting us.

REFERENCES

[1] DUC. Document understanding conference 2002, 2002. http://www.nlpir.nist.gov/ projects/duc

[2] Fattah M.A. and Ren F. 2008. Automatic Text Summarization, *In proceedings of World Academy of Science, Engineering and Technology* Volume 27.192-195.

[3] Kiani A. and Akbarzadeh, M.R. 2006. Automatic Text Summarization Using: Hybrid Fuzzy GA-GP. *In Proceedings of 2006 IEEE International Conference on Fuzzy Systems*, Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada. 977-983.

[4] Ko Y., Seo J. 2008. An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. *Pattern Recognition Letters archive*, vol. 29, Issue 9, pp. 1366-1371. DOI: 10.1016/j.patrec.2008.02.008

[5] Luhn H. P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, vol. 2. 159-165.

[6] Kupiec J., Pedersen, J. and Chen, F. 1995. A Trainable Document Summarizer. *In Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, Seattle, WA, 68-73.

[7] Lin C.Y. 1999. Training a selection function for extraction. *In Proceedings of the eighth international conference on Information and knowledge management*, Kansas City, Missouri, United States. 55-62.

[8] Lin C.Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. *In Proceedings of Workshop on Text Summarization of ACL*, .Spain.

[9] Mani I. and Mark T. Maybury, (editors). 1999. *Advances in automatic text summarization* MIT Press.

[10] Morris G., Kasper G.M., and Adam D.A. 1992. The effect and limitation of automated text condensing on reading comprehension performance. *Information System Research*, 3(1), 17-35.

[11] Porter M. F., 1980. An algorithm for suffix stripping. *Morgan Kaufmann Multimedia Information And Systems Series.*, Morgan Kaufmann Publishers Inc. 313-316. ISBN: 1-55860-454-5

[12] Ramiz M. Aliruliyev, 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems*

with Applications: An International Journal archive 36(4). 7764-7772. DOI: 10.1016/j.eswa.2008.11.022

[13] Salton G. 1989. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Publishing Company.

[14] Salton G. and Buckley C. 1997. Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*, 1988. 513-523. Reprinted in: Sparck-Jones, K.; Willet, P. (eds.) Readings in I.Retrieval. Morgan Kaufmann. 323-328.

[15] Srinivas M. and Patnaik L. M., 1994. Genetic algorithms: A Survey. *Computer* 27(6) (Jun. 1994), 17-26. DOI= http://dx.doi.org/10.1109/2.294849

[16] Suanmali, L., N. Salim and M.S. Binwahlan, 2009. Fuzzy Logic Based Method for Improving Text Summarization. *International Journal of Computer Science and Information Security (IJCSIS),* vol. 2(1), 65-70. http://arxiv.org/abs/0906.4690

[17] Vahed, Q., Leila Sharif, H., and Ramin, H.,2008. Summarising text with a genetic algorithm-based sentence extraction. *International Journal of Knowledge Management Studies*, vol. 2(4), 426-444.

[18] Yeh, J.Y., H.R. Ke, W.P. Yang and I.H. Meng, 2005. Text summarization using a trainable summarizer and latent semantic analysis. *In: Special issue of Information Processing and Management on An Asian digital libraries perspective*, 41(1), 75–95.

[19] Yulia, L., G. Alexander, and René Arnulfo García-Hernández, 2008. Terms Derived from Frequent Sequences for Extractive Text Summarization. *In: A. Gelbukh (Ed.): CICLing 2008, LNCS* vol. 4919, Springer, Heidelberg, 593-604. DOI: 10.1007/978-3-540-78135-6

Mrs. Ladda Suanmali is a Ph.D. candidate in computer science in the Faculty of Computer Science and Information Systems at Universiti Teknologi Malaysia. She received her B.Sc. degree in computer science from Suan Dusit Rajabhat University, Thailand in 1998, and her M.Sc. degree in Information Technology from King Mongkut's University of Technology Thonburi, Thailand in 2003. Since 2003, she has been working as a lecturer at Suan Dusit Rajabhat University. Her current research interests include text summarization, data mining, and soft computing.

Dr. Naomie Salim is an Assoc.Prof. presently working as a Deputy Dean of Postgraduate Studies in the Faculty of Computer Science and Information System in Universiti Teknologi Malaysia. She received her degree in Computer Science from Universiti Teknologi Malaysia in 1989. She received her Master degree from University of Illinois and Ph.D Degree from University of Sheffield in 1992 and 2002 respectively. Her current research interest includes Information Retrieval, Distributed Database and Chemoinformatic.

Dr. Mohammed Salem Binwahlan received his B.Sc. degree in Computer Science from Hadhramout University of Science and Technology, Yemen in 2000. He received his Master degree and Ph.D Degree from Universiti Teknologi Malaysia in 2006 and 2011 respectively. His current research interest includes Information Retrieval, Text Summarization and Plagiarism Detection.