Int. J. Advance Soft Compu. Appl, Vol. 7, No. 1, March 2015
ISSN 2074-8523

The Feature Selection Method based on Genetic Algorithm for Efficient of Text Clustering and Text Classification

Sung-Sam Hong[1], Wanhee Lee[2], and Myung-Mook Han[1]*

[1]Department of Computer Engineering, Gachon University
e-mail:sungsamhong@gmail.com, mmhan@gachon.ac.kr
[2]Department of Police Science & Security Studies
e-mail:davincidecoder@gmail.com
*corresponding author

Abstract

Big Data means a very large amount of data and includes a range of methodologies such as big data collection, processing, storage, management, and analysis. Since Big Data Text Mining extracts a lot of features and data, clustering and classification can result in high computational complexity and the low reliability of the analysis results. In particular, a TDM (Term Document Matrix) obtained through text mining represents term-document features but features a sparse matrix. In this paper, the study focuses on selecting a set of optimized features from the corpus. A Genetic Algorithm (GA) is used to extract terms (features) as desired according to term importance calculated by the equation found. The study revolves around feature selection method to lower computational complexity and to increase analytical performance.We designed a new genetic algorithm to extract features in text mining. TF-IDF is used to reflect document-term relationships in feature extraction. Through the repetitive process, features are selected as many as the predetermined number. We have conducted clustering experiments on a set of spam-mail documents to verify and to improve feature selection performance. And we found that the proposal FSGA algorithm shown better performance of Text Clustering and Classification than using all of features.

Keywords: *Big Data, Text Mining, Genetic Algorithm, Text Clustering, Feature Selection.*

1 Introduction

This Big Data means a very large amount of data and includes a range of methodologies such as big data collection, processing, storage, management, and analysis. In particular, text mining among unstructured big data, which is recently utilized in many industries, is an important unstructured data analysis technique. Text mining is likely to extract a larger number of terms (features) as the amount of data get larger. Since Big Data Text Mining extracts a lot of features and data, clustering and classification can result in high computational complexity and the low reliability of the analysis results. In particular, a TDM(Term Document Matrix)[3] obtained through text mining represents term-document features but features a sparse matrix. In the case of a sparse matrix, useful information cannot be retrieved and the analysis result cannot be trusted. Therefore, there have been various studies on feature selection and data dimension [2, 3, 4].

In this paper, the study focuses on selecting a set of optimized features from the corpus. A Genetic Algorithm(GA) is used to extract terms (features) as desired according to term importance calculated by the equation found. The study revolves around feature selection method to lower computational complexity and to increase analytical performance. The Genetic Algorithm[1] is used to find the optimum in order to solve arrangement and assignment problems. We have done research on a variant of genetic algorithm which is used to find the optimal term group by computing the fitness of terms. We designed a new genetic algorithm to extract features in text mining. TF-IDF[16] is used to reflect document-term relationships in feature extraction. Through the repetitive process, features are selected as many as the predetermined number. We have conducted clustering experiments on a set of spam-mail documents to verify and to improve feature selection performance. We have also verified its performance by applying the proposed algorithm to text clustering and classification.

The rest of the paper is organized as follows. In Section 2, related work is introduced. In Section 3, feature selection techniques using genetic algorithms in text mining are described. In Section 4, the experiment and analysis results are presented. Finally, the conclusions are described in Section 5.

2 Related Works

2.1 Text Mining

The *Text Mining*[3] can be referred as a data mining technology which derives valuable and meaningful information from unstructured text data. The text mining technology allows users to extract meaningful information from a vast amount of information and to identify the relationships with the other information. It also involves text categorization, simple information retrieval, etc. High-capacity

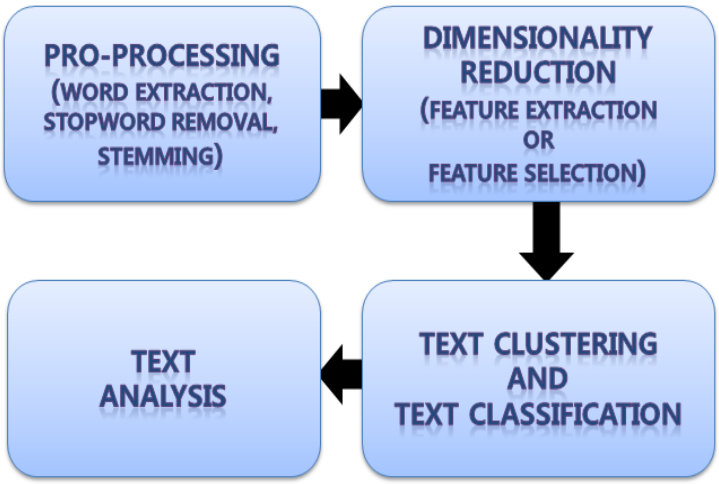

Fig. 1 General Processing of Text Mining

linguistic resources and complex statistical pattern learning algorithms are used to make the computer do an in-depth analysis of the information written in human language and to discover the hidden information from the given information. As data become big data and social networks including SNS and blogs are expanded, text Mining is widely being used for advertising, marketing, law case analysis, information retrieval, trend analysis, etc.

Text Classification uses data classification algorithms which are commonly used for in the existing data mining. Data classification uses the predetermined classification criteria to learn data through class-specific learning and then uses the result of learning to classify the input data into predetermined classes. Unlike conventional text classification, *Text Clustering* doesn`t use the predetermined criteria but analyzes the given data to group similar data into the same cluster according to the features of the data such as similarity. Accordingly, it is possible to show the characteristics of the given data and to extract the unknown data hidden knowledge. That is, it is possible to make a prediction of the given data. As shown in Fig.1 [15], the basic process of text mining is as follows. The most basic unit, corpus, refers to a set of text documents, and the corpus pre-processing is the first step. Natural language processing techniques are mainly used for word extraction, stop word removal and stemming tasks. Next, feature extraction and feature selection are carried out for dimension reduction of the term matrix, and then text classification and text clustering proceed with the created feature set. Finally, text analysis is performed.

2.2 Genetic Algorithm

Genetic Algorithm is a major *Heuristic Algorithm* which mimics Darwin's theory of evolution, and it is an *Evolutionary Algorithm* which finds the optimal solution in the process of natural selection and crossover [1]. As shown in Fig.2, the process of genetic algorithms is as follows. Genetic Algorithms randomly generates initial individuals to form an initial population. Each individual consists of a variable *Gene* that represents a solution to the given problem and encoded by *Chromosome*. Depending on the specific optimization problems, different encoding methods are applied. For example, the binary or real-valued representation can be used to express schemata. The representation selection can have a significant impact on the successful application of Genetic Algorithms. An *object function* is designed to express the problem in *Genetic Algorithm*s and the *fitness value* is obtained by applying each individual to the object.

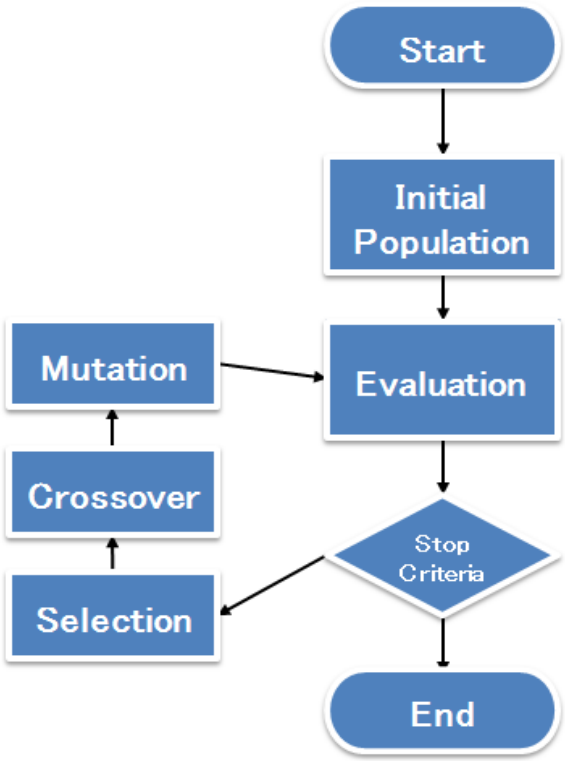

Fig. 2 Genetic Algorithm

Genetic Algorithm design is to include the following three important operators: *Selection, Crossover* and *Mutation*. The selection operator is the process of selecting individuals the next generation from the current generation. The selection operator is usually designed to select probabilistically good solutions (individuals with high Fitness Values) and remove other bad solutions. The crossover operator is the process of gene recombination which recombines two parents' chromosomes

to generate new individuals to be used in the next generation. For example, there are the following major crossover techniques: one-point crossover and two-point crossover. The mutation operator is the process of altering one or more gene values randomly selected in the current chromosome. The generational process based on genetic operators is repeated to gradually evolve candidate solutions which converge on approximate solutions more and more. When the *Genetic Algorithm* process is terminated due to the given constraints, the optimum of the solutions is obtained to solve the problem. *Genetic Algorithms* have been used in many areas including *TSP(Traveling Salesman Problem),job scheduling, network layout, channel routing, graph partitioning, DB(Database), query optimization, etc*.

Likewise, *Genetic Algorithms* have been used for malware detection and intrusion detection in the field of information protection so that there have been studies on *Intelligent Intrusion Detection System*[8].

3 FSGA : Feature Selection based on Genetic Algorithm

In order to apply a Genetic Algorithm to feature selection, it is necessary to design the Genetic Algorithm to meet its given domain [5]. In this paper, a new *Genetic Algorithm* for feature selection is designed to improve the analytical performance and speed in text mining step-by-step.

3.1 Initial Population

The initial population is formed from the TDM generated in the process of text mining. The rows (terms) and columns (documents) in the TDM are converted into the rows (documents) and columns (terms) in the DTM (Document-Term Matrix), where each column (term) represent a feature and each row (document) represents a solution. This vector matrix is used to set the initial population.

In text mining, many features are generated unlike the existing general data set. Given the regular mail text used in this paper, about 4000 features are generated from 100 documents. About 11,000 features have been generated from 300 documents used in the experiment. Therefore, it is too difficult to represent all features of the initial population as chromosomes in the search space. So we have randomly generated chromosomes of fixed length according to the population size. The single *GA* process isn`t carried. That is, once the Optimum is obtained, the GA process is repeated until the desired feature can obtained. When the multiple GA processes are performed, various problem domains can be explored and genetic diversity can be ensured.

3.2 Chromosome Decoding

Chromosomes in Genetic Algorithms are used to represent solutions, and chromosomes are decoded in the GA process, depending on the properties of domains. We have designed chromosomes for term feature selection in text mining

as follows, and each solution can be classified by one partial feature, and many partial features is gathered to form a feature set.

• Permutation Decoding = the permutation of the gene values is expressed to represent the terms

• The selected partial features are stored in the feature set

For example, the chromosome *C* is expressed such as *C=(work, speak, will, paper,...)* and is randomly selected in the initial population step. In order to perform the actual GA process, the permutation of terms is to be expressed in TDM, where index masking is applied to each term.

3.3 Fitness Function Studies and Design

For example The fitness function in *Genetic Algorithms* is the fitness equation which is used to evaluate the superiority of the given solution. In this study, the fitness function for text mining has been designed to evaluate the importance of the given term. The corresponding notation is as follows

• $F = \{F_1, F_2,\ldots,F_n\}$ ≜the set of Features (Chromosome)

• $D = \{D_1, D_2, \ldots, D_n\}$≜the set of Documents

• N = the number of documents

• $x_i = F_i \triangleq$ value of ith Feature in $\in F$

• tf_{ik}≜term frequency of feature $F_i \in F$ in document $D_k \in$D

• df_k≜document frequency is the number of document included $F_k \in F$

• idf_k≜inverse document frequency of feature $F_k \in F$ in document $D_k \in D$

$= \log((N - df_k)/df_k)$

Definition 3.1 *Fitness Function*

$$max\ F = \sum_{i=1}^{|F|} \sum_{k=1}^{|D|} (tf_{x_i k} \times idf_k)$$

Regarding the importance of the given term, the Definition 3.1 doesn`t simply apply its overall frequency, but it uses its relative frequency idf_k with respect to each document and term to obtain the fitness value of each solution. Therefore, the relative importance can select the feature. When the frequency of the term is

simply used, the term can occur frequently but its meaning can be used differently in each document. However, this approach is not suitable for the low-frequency term which may represent the important feature of documents. Therefore, the fitness function using TF-IDF has been designed in consideration of the Document-Term relationships.

3.4 Selection and Crossover Operator (in GA)

For example In *Genetic Algorithms*, there are many selection techniques which select individual chromosomes with high fitness. The selected chromosome is used by the crossover operator. The superior chromosome is chosen to propagate its superior gene to the next generation. That is, superior genes can be propagated to good solutions and bad solutions to ensure genetic diversity. We have used the *linear rank selection* method[7].

Crossover is a genetic operator used to recombine two parents chromosomes to generate new individuals in the next generation. Crossover aims at increasing genetic diversity just like selection. We have used the *position-based crossover*method[6].

3.5 Feature Selection

For example A large number of features are mostly generated by using terms extracted from the given corpus. In *Genetic Algorithms*, it is too difficult to perform feature selection due to its runtime complexity and iteration. Therefore, we haven't done feature selection in the single GA process. That is, we have generated the partial features of fixed length through the GA process and grouped each partial feature to obtain a final feature set. When the multiple GA processes are performed, various problem domains can be explored and genetic diversity can be ensured. The following Table 1 shows the feature selection process. Since the process is repeated multiple times, the same features can be selected again. The feature with a high frequency can be very important so that the repetitive process is allowed. The length of the final feature set is determined in proportion to the fixed length of features generated from the corpus.

A large number of features are mostly generated by using terms extracted from the given corpus. In *Genetic Algorithms*, it is too difficult to perform feature selection due to its runtime complexity and iteration. Therefore, we haven't done feature selection in the single GA process. That is, we have generated the partial features of fixed length through the GA process and grouped each partial feature to obtain a final feature set. When the multiple GA processes are performed, various problem domains can be explored and genetic diversity can be ensured. The following Table1 shows the feature selection process.

Table 1: Procedure of Feature Selection

Algorithm : Feature Selection Procedure

tdmFeaLength : The number of features in tdm
lSubFeature : The number of sub feature
lFinalFeature : The number of final feature set
maxGen : Maximum generation
limitTime : Limit of elapsed Time
popsize : Population Size

```
// length of chromosome = iSubFeature
// value of gene = index of feature(1 < index < tdmFeaLength)

while (lFinalFeautre > length of finalFeautre) {
   p <-initialPopulation(tdmFeaLength, iSubFeature, popsize)
   computeFitness(p)
   generation = 1
   while(maxGen >= generation OR limitTime > ElapsedTime) {
      newPop <- linearRankSelection(p)
      pbxCrossover(newPop)
      mutate(newPop)
      p <- newPop
      computeFitness(p)
    ElapsedTime = ElapsedTime+ nowTime
      generation = generation + 1
   }
   subFeature <- TopFitnessFeature(p)
   finalFeautre <- addFeature(subFeature)
}
return finalFeature
```

4 Experiment and Analysis

4.1 Experiment Environment

The hardware and operating system environment used in the experiment is as follows.

- CPU : Intel Core i5 650 3.20Ghz
- RAM : 7GB
- OS : Windows 7 Enterprise K 64bit

The open software *R(*version 3.02) [9] is a tool that has been used for the experiment. In particular, the *TM(TextMining)* package in R[10] has been used for text mining. GA algorithms in the R GA package [11] have been modified run them in the R environment. Clustering algorithms used for the clustering experiments are *K-means Algorithm*[12] and *HierarchicalClustering* [13]. Classification Algorithm used for the classification experiments is KNN Classifier[19].

4.2 Document Data Set

The document data set is used in the experiments are 300 documents from *LingSpam Data Set*[14] which is used for spam mail classification and clustering. The dataset has been classified into two clusters: normal mail and spam mail. The clustering performance has been measured. These experiments have used. 252 normal mails and 48 spam mails.

4.3 Experiment Result

The feature selection experiment has been performed by using the TF-IDF GA. The clustering results have been analyzed and compared with the original corpus. The GA parameters used are as follows:

- Size of population = 200
- length of chromosome = 55
- probability of crossover = 0.8
- probability of mutation = 0.2

4.3.1 Measure Method of Clustering Experiment Result

The unsupervised clustering performance is measured as follows [15].

- *ss*: in our clusters and in the corpus both documents are placed in the same clusters.
- *sd*: in our clusters both documents are placed in the same clusters but in corpus are in different clusters.
- *ds*: in our clusters documents placed in different clusters but in the corpus are in the same clusters.
- *dd*: in our clusters and in the corpus both documents placed in different clusters.

We use the measure method that the Average Accuracy and F1-measure. The average accuracy for Clustering is defined as follows:

Definition 4.1 $Average\ Accuracy(AA)$

$$= \frac{1}{2} \times \left(\frac{ss}{ss + ds} + \frac{dd}{sd + dd} \right)$$

The F1-measure for Clustering is defined as follows:

Definition 4.2 *F1-measure*

$$F1 - measure = \frac{2 \times p \times r}{p + r}$$

where

$$p = \frac{ss}{ss + sd}, r = \frac{ss}{ss + ds}$$

For these experiments, the length of the sub-feature set used is 55, and the length of the final feature set has been set to about 5%, 10%, 15%, 20% and 50 % of the total number of the TDM features respectively. 11,508 features have been generated from 300 documents. Given the features, 550 features, 1,100 features, 1,650 features, 2,200 features, and 5,500 features have been selected. According to the length of each final feature set, the AA and F1-measure values have been measured by using each clustering algorithm.

4.3.2 Clustering Experiment Result

The overall clustering result is as shown in Table.2. The % values represent the ratio of the length of each final feature set selected from the total features.

Given the average accuracy results (Fig.3), *hClust(Hierarchical Clustering)*has shown the low *average accuracy(AA)* from 5% to 0.795, and its AA is increasing as the length of the feature set (the number of features increases) gets longer. In particular, the performance of *hClust* gets better than that of K-means, where the ratio of the feature set is 20% and 50% and AA is is 0.936 and 0.941 respectively. In addition, it can be known that the clustering performance gets better when the ratio of the length of the feature set is greater than or equal to 15% in comparison with all of features. K-means has shown the stably good AA performance for the *ratio of each feature set.* In particular, it shows better performance than clustering

using all features. Thus, it is can be seen that each clustering Algorithm with *FSGA(Feature Selection based on Genetic Algorithm)* can improve the AA performance, compared to the clustering Algorithm without FSGA .

Table 2: A Result of Experiment

Average Accuracy						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
hClust	0.739	0.819	0.930	0.94	0.945	0.925
K-means	0.938	0.916	0.929	0.926	0.926	0.821
F1-Measure						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
hClust	0.795	0.939	0.939	0.936	0.941	0.918
K-means	0.934	0.936	0.938	0.934	0.934	0.976

Table 3: A Result of Clustering Time

Clustering Time(s)						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
hClust	0.22	0.46	1.01	1.92	6.36	13.84
K-means	0.01	0.03	0.11	0.1	0.26	0.49

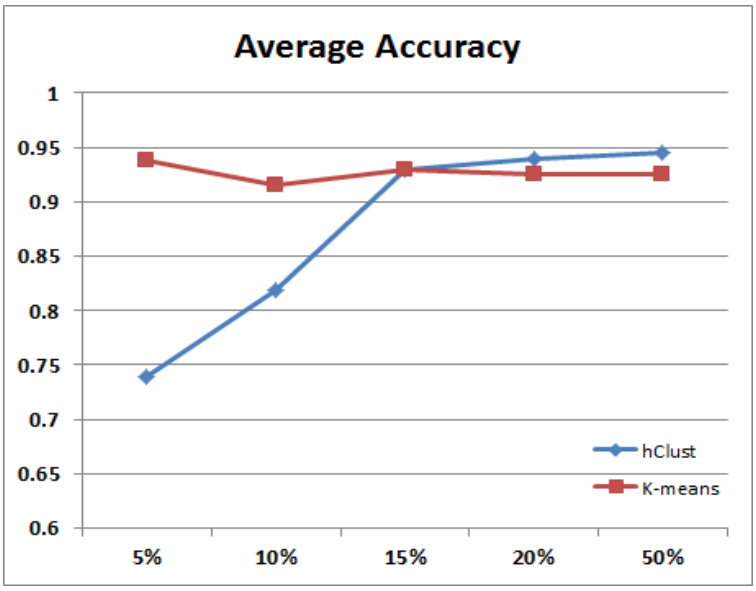

Fig. 3 A Result of Average Accuracy

Given the F1-measure results (Fig.4), hClust has shown low performance (0.795) for 5%. When F1-measure becomes similar to *k-means* for10%, it has shown better performance by a narrow margin of 0.003~0.007. hClust has shown the performance of 0.918 for all the feature sets, and it has also shown high performance when the ratio of the feature set is greater than or equal to 10% in comparison with all of features. *The* k-means's F1-measures for all the feature sets turn out to be lower than 0.976 which indicates the F1-measures for all the features. However, *k-means' average* F1-measure is 0.935 which indicates good performance. As a result of AA and F1-Measure experiments, it can be know that FSGA can lead both hClust and k-means to improve the AA performance, and *FSGA* can also lead hClust to improve the F1-measure performance.

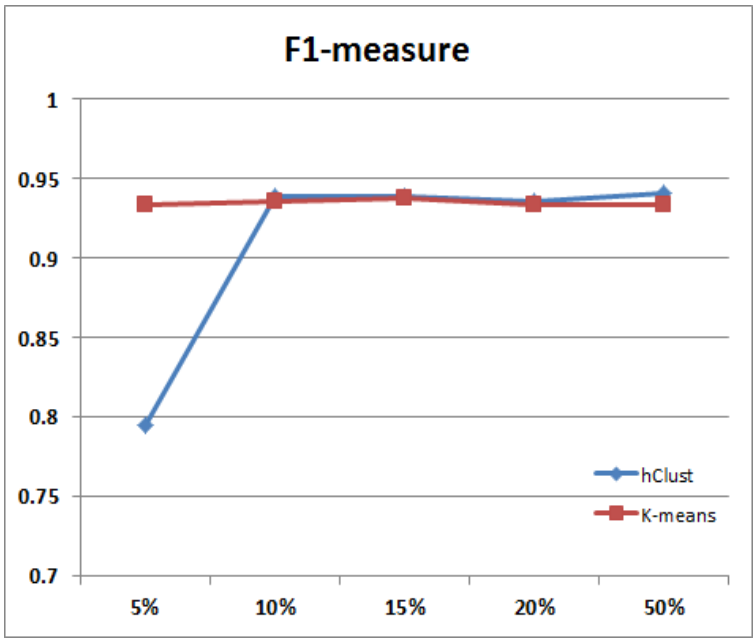

Fig. 4 A Result of F1-measure

4.3.3 Clustering Time Result

The required clustering time has been measured through experiments. The experiment results are shown in Table.3. In all experiments, both *hClust* and *k-means* have shown that the clustering time for each feature set is shorter than the clustering time for all of features. The speed of *k-means* turns out to be faster than that of *hClust*. As shown in Fig.5, *All of Features* and *FSGA* have a great impact on the speed of *hClust*. These These experimental results indicate that the clustering time decreases as the length of features get shorter.

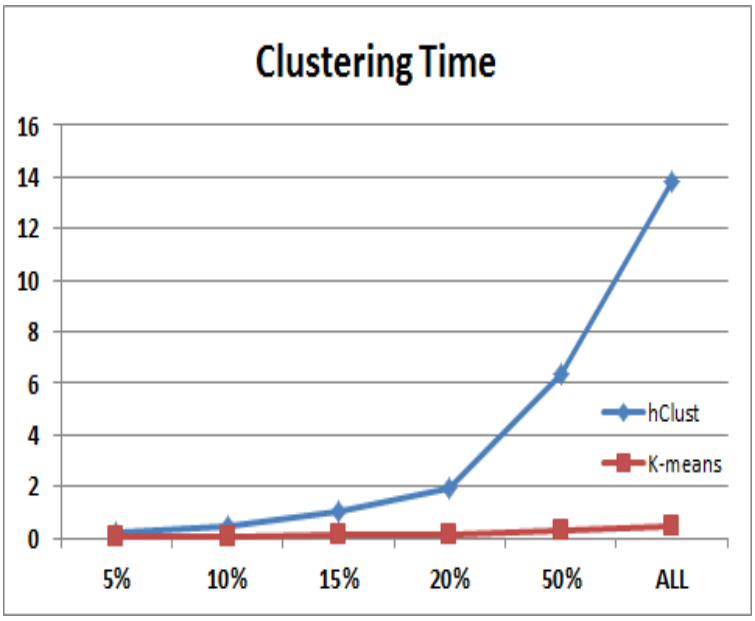

Fig. 5 A Result of Clustering Time

4.3.4 Measure Method of Text Classification Experiment

These are In the evaluation step, the score function is applied to evaluate the performance by using the feature set derived from the feature selection method. Heuristic Search is repeatedly used to find the desired feature set to meet the given criteria in the learning process until the final feature set is selected. The optimal feature set is to be chosen by using evaluation and search because a partial feature set is selected from the high-dimensional feature sets.

In this case, the F-Measure is used for the score regarding document classification evaluation. The F-measure is widely used to evaluate the results of text classification by applying precision and recall [18]. The precision Pi and the recall Ri is calculated by the following Definition 4.3, 4.4

Definition 4.3 *Precision*

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

Definition 4.3 *Recall*

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

where TP*i*, FP*i* and FN*i* stand for true positives, false positives and false negatives respectively. The F-measure for the category*i* is calculated by the following Definition 4.5.

Definition 4.5 *F-measure*

$$F_i = \frac{2 * P_i * R_i}{P_i + R_i}$$

The average F-measure is calculated by the following Definition 4.6,

Definition 4.6 *Average F-measure*

$$F_i = \frac{\sum_{i=1}^{N} d_i \cdot F_i}{\sum_{i=1}^{N} d_i}$$

where di indicates the number of the documents included in the category i. N is the number of categories.

4.3.5 Classification Experiment Result

In the experiment, Precision, Recall, F1-measure values were calculated for each feature selection ratio. In the experiment, the average value of classification was obtained by carrying out 10 experiments for each feature selection ratio. The result of the classification experiment is shown in Table 4. The value in each parenthesis is the standard deviation of each experiment.

The result of the experiment shows that, in the case where FSGA was carried out, the best results were achieved when only 5 % was selected for Precision, 50 % for Recall and when 50 % was selected for F-measure showing the values of 0. 923, 0.995 and 0.948 respectively. When the results are compared with those achieved by using all the features, the best performance was achieved when all the features were used in the cases of F-measure and Precision. On the other hand, better result

Table 4: A Result of Classification Experiment

Precision						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
KNN	0.923	0.897	0.906	0.910	0.906	0.983
(stdev)	(0.027)	(0.019)	(0.018)	(0.022)	(0.016)	-
Recall						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
KNN	0.951	0.977	0.960	0.989	0.995	0.967
(stdev)	(0.022)	(0.058)	(0.088)	(0.016)	(0.016)	-
F1-Measure						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
KNN	0.936	0.934	0.929	0.948	0.948	0.975
(stdev)	(0.016)	(0.036)	(0.050)	(0.013)	(0.010)	-

was achieved when FSGA was used in the case of Recall. Analysis of the overall experiment result shows that the higher the feature selection ratio is, the more the values of Precsion, F-measure, and Recall improve in general with the exception of several cases as shown in Fig 6. In the case of Classification, as learning is carried out through already classified criteria differently from Clustering, it can be seen that the bigger the data used for learning is, the more the classification result

improves. However, as the classification performance did not deteriorate very much even when FSGA was used and superior result was achieved depending on the selection ratio, superior performance and utilization potential of FSGA could be verified also in document classification. However, in order to show better classification performance, the performance of FSGA is required to be enhanced.

Table 5 shows the experiment result of Classification time for each feature selection ratio. Fig 7 shows the result in a graph. It can be seen that, the higher the feature selection ratio becomes, the more the performance time greatly inceases. Accordingly, when we see the previous result of classification performance experiment and the result of performance time, document classification using FSGA can be valuably used depending on the purpose of use or the environment.

Table. 5 A Result of Classification Time

Classfication Time(s)						
	5%	**10%**	**15%**	**20%**	**50%**	**ALL**
KNN	0.02	0.09	0.14	0.23	0.61	1.34

As the classification time can be greatly reduced while maintaining the document classification performance to some extent within the usable level, it can be valuably utilized under the environment of big data, real-time document classification, or where computing resources are insufficient. If the purose is to accurately carry out classification irrespective of the document classification time, a better result may be achieved by using all the features.

When the overall result of the document classification experiment is considered, the proposed FSGA algorithm can be said to be a method that can be efficiently utilized for document classification.

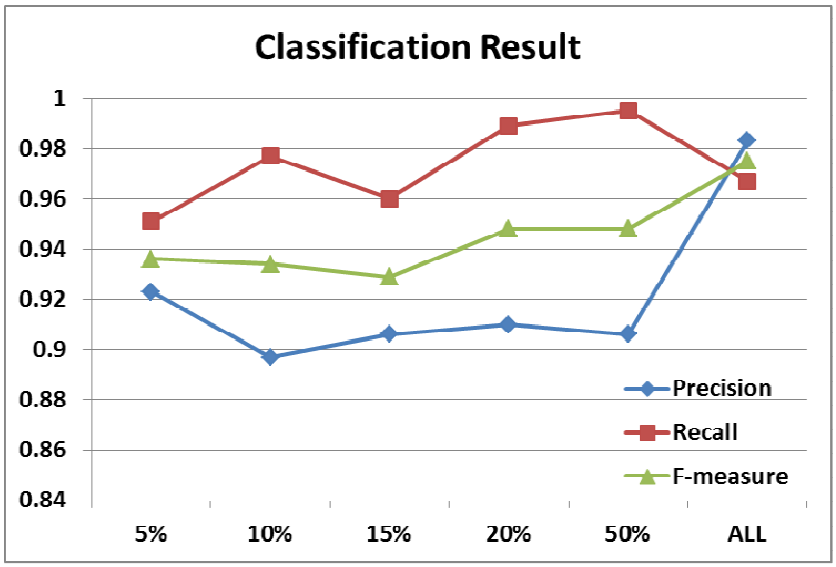

Fig 5. A Result of Classification Experiment

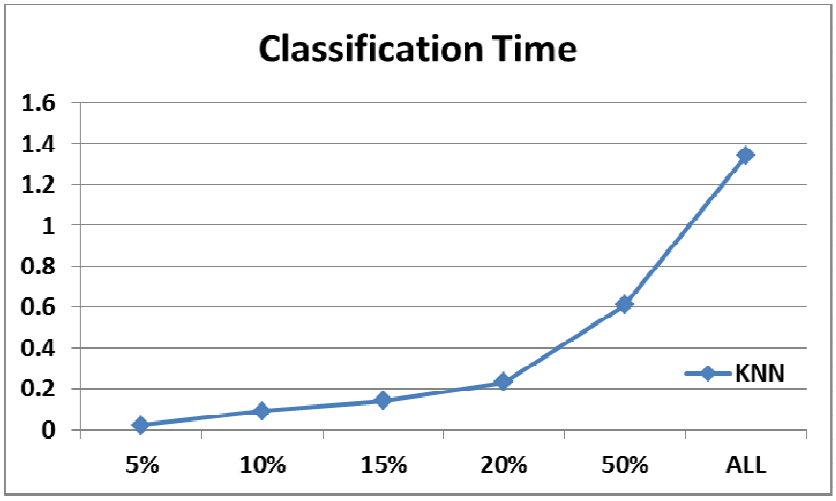

Fig 6. Classification Time

5 Conclusion

In this paper, a feature selection (Term Selection) method is proposed to enhance the effectiveness of the analysis in text mining. A new *Genetic Algorithm* has been designed to be applied to text mining due to it optimum search performance. In addition, in order to keep genetic diversity, the algorithm has been modified to select the final feature set by using partial feature sets. Also, the FSGA performance has been verified through hierarchical clustering and k-means experiments. Regarding the hierarchical clustering algorithm, the Average Accuracy and F1-measure of clustering using FSGA has been improved up to 0.02 and up to 0.023 respectively. Regarding the k-means algorithm, the Average Accuracy of clustering using FSGA has been improved up to 0.105, but the F1-measure of clustering using FSGA has been lowered. Using FSGA has shown high speed performance in all cases. As a result, it can be known that using the FSGA can improve clustering performance and performance speed.

In the document classification experiment, the classification performance was shown to be better when all features were used than when FSGA is used with an exception of the result of Recall. Such a result is presumed to have been achieved as, the bigger the number of the objects of learning is, the higher the classification performance can be, because learning is used for classification. However, even when FSGA was used, the classification performance was not shown to be very much lower, and the results of Precsion, Recall, and F-measure showed document classification performance of a usable level. As the performance time required for classification can be relatively reduced a lot, it can be efficiently utilized for document classification under the environment of real-time data processing or where computing resources are insufficient.

For future research, there will be studies on the new GA design using *Parallel Genetic Algorithm* to improve the performance of GA repetition. It is also necessary to improve the proposed algorithm regarding duplicated feature selection and fitness function. Furthermore, there will be studies on FSGA suitable for the *Big Data* environment to deal with large amounts of text document data. And we will carry out studies on the Text Clustering based on Optimization Algorithms[17]. We will also study the method that can improve the FSGA performance in document classification.

ACKNOWLEDGEMENTS
This research was funded by the MSIP(Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2014.

References

[1] J. H. Holland, 1989, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison- Wesley, Reading, MA

[2] Manoranjan Dash and Huan Liu, 2000, “Feature Selection for Clustering,” *IN PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*

[3] Hearst and Marti A. , 1999, “Untangling text data mining,” *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 3–10

[4] I. S. Dhillon, Y. Guan and J. Kogan, 2002, “Refining clusters in high-dimensional text data,” *Proceedings of the Workshop on Clustering High Dimensional Data and its Applications at the Second SIAM International Conference on Data Mining*, pp. 71–82.

[5] Il-Seok Oh, Jin-Seon Lee and Byung-Ro Moon, 2004, “Hybrid Genetic Algorithms for Feature Selection,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 26, NO. 11

[6] I.M.Oliver, D.J.Smith , J.R.C Holland, 1987, “A Study of Permutation Crossover Operator on the TSP,”*Proceedings of the Second International Conference*, pp224-230

[7] Michalewicz, Z. , 1996, “Genetic AlgorithmsData StructuresEvolution Programs” New-York: Springer Verlag. 3rd edition

[8] Ji-Hong Yang, Myung-Mook Han, 2002, “The Intelligent Intrusion Detection Systems using Automatic Rule-Based Method,” *Korean Institute of Intelligent System*, Vol12, No.6, pp. 531-536

[9] http://www.r-project.org/

[10] http://cran.r-project.org/web/packages/tm/index.html

[11] http://cran.r-project.org/web/packages/GA/vignettes/gaJSS.pdf

[12] MacQueen, J. B. , 2009 ,"Some Methods for classification and Analysis of Multivariate Observations". *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1*. pp. 281–297.

[13] R. Sibson, 1973, "SLINK: an optimally efficient algorithm for the single-link cluster method". *The Computer Journal (British Computer Society)*,Vol16, No.1, pp 30–34

[14] Androutsopoulos, 2000., J. Koutsias, K.V. Chandrinos, George Paliouras, and C.D. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering", *11th European Conference on Machine Learning (ECML 2000),* pp. 9-17

[15] PiroozShamsinejadbabki ·Mohammad Saraee, 2012, “A new unsupervised feature selection method for text clustering based on genetic algorithms,” The Journal of Information System, Vol 38, pp 669-684

[16] Robertson, Stephen, 2004, "Understanding inverse document frequency: On theoretical arguments for IDF". Journal of Documentation, Vol.60, No.5, pp.503–520

[17] R.Jensi,Dr.G.WiselinJiji, 2013, “A SURVEY ON OPTIMIZATION APPROACHES TO TEXT DOCUMENT CLUSTERING,” *International Journal on Computational Sciences & Applications (IJCSA)*, Vol.3, No.6, pp.31-44

[18] J. Van Rijsbergen, 1979, Information Retrieval, second ed., *Buttersworth*, London

[19] Altman, N. S, 1992, "An introduction to kernel and nearest-neighbor nonparametric regression", *The American Statistician*, Vol.46, No.3, PP.175–185.