International Journal of Grid and Distributed Computing
Vol. 11, No. 4 (2018), pp.97-112
http://dx.doi.org/10.14257/ijgdc.2018.11.4.09

Document Classification Method based on Latent Semantic Indexing

Jeong-Joon Kim[1], Yong-Soo Lee[2], Jin-Yong Moon[3] and Jeong-Min Park[4*]

[1]Korea Polytechinc University, Gyeonggi-do, Korea
[2]Yeoju Institute of Technology, Gyeonggi-do, Korea
[3]Gangdong College, Gyeonggi-do, Korea
[4]Korea Polytechinc University, Gyeonggi-do, Korea
[1]jjkim@kpu.ac.kr, [2]diclee@yit.ac.kr, [3]jmoon37@gmail.com, [4]jmpark@kpu.ac.kr

Abstract

Among the studies, Latent Semantic Indexing and Non-negative Matrix Factorization, which are algorithms to classify the document by meaning, try solve the problems by converting the document to vector. However, there are 2 problems in these algorithms that the different understanding according to education document and the difficulties to analyze the multiple representations of the terms. Meanwhile, WordNet is a word dictionary interpreting the relationship of the words based on Human Intelligence Science and widely used in such as query term extension of the search engine. However, it is difficult to adapt to the neologism and slang and word meaning change to fast-changing time.

Therefore, in this paper we solve the problem of the multiple representations of the words by partly applying the words relationship of the WordNet to Latent Semantic Indexing using by genetic algorithms for more efficient clustering document with the strength and weakness of the Latent Semantic Indexing and WordNet. And with this we try to improve precision and increase the efficiency of the overall clusters

Keywords: *Document Clustering, WordNet, Latent Semantic Indexing, Genetic Algorithm, CRSS*

1. Introduction

The document classification shows that it is similar to algorithms in many fields of information retrieval such as clustering, search, ranking, etc. It is possible to classify documents using algorithms in the field of information retrieval[1,2]. There is a potential semantic analysis(Latent Semantic Analysis) method by a representative method. Potential meaning analysis method was announced in the mid 1950s, but along with the development of information retrieval technology, it became like soybeans in the mid 1990 's. After that, many researches were carried out based on the latent meaning analysis method. However, the potential meaning analysis method can't grasp the ambiguity of the vocabulary. As a vocabulary reference system created by the Cognitive Science Institute of Princeton University in 1985, WordNet is sometimes classified as a database that summarizes the vocabulary system and phase in a psycholinguistic theory about human vocabulary concepts[3,4].

The Genetic Algorithm is a global optimization method devised by John Holland which is a calculation model based on the process of evolution of the natural world, expressed in a data structure defined as possible with parallel global navigation While gradually deforming it, it is approaching more and more nice. It is a method for obtaining

Received (October 19, 2017), Review Result (January 20, 2018), Accepted (January 25, 2018)
* Corresponding Author

ISSN: 2005-4262 IJGDC

an answer that is close to the optimum solution although it does not obtain an actual optimal solution [5,6]. In the GAOLSF (Genetic Algorithm Oriented Latent Semantic Features), as a paper attempting to select efficient qualities using a genetic algorithm, probabilistically candidates are generated to have better result values did[7,8]. However, we can't grasp the ambiguity of the vocabulary. WordNet clustering based on fuzzy rule is done by strengthening the document using wordnet 's parent word and then constructing clusters by constructing document cluster matrix using fuzzy rules[9]. However, there may be inaccurate results depending on the parent word relationship in the document enhancement process.

In this paper, we aim at the vagueness ambiguity resolution and high quality clustering result using wordnet, latent semantic index, genetic algorithm, and for that purpose utilize CRSS (cosine residual sum of squares) Genetic algorithm was evaluated.

2. Related Work

2.1. Genetic Algorithm Oriented Latent Semantic Features

Genetic Algorithm Oriented Latent Semantic Features (GAOLSF) is a paper that efficiently selects the qualities using Genetic Algorithm. The feature selection is a task of selecting the important lexicon used to classify the document and eliminating the low weighted words, which can reduce the noise, reduce the calculation and improve the accuracy. Genetic Algorithm is used to regenerate a good quality document by using regression. GAOLSF generates candidate solutions in two ways: Distinguishing Feature Selector (DFS) and Chi Square (CHI2). Table 1 shows the expressions of DFS and CHI2.

Table 1. DFS, CHI2

DFS(Distinguishing Feature Selector)

- $DFS(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + P(t|\overline{C_i}) + 1}$

M : Number of classes

$P(C_i|t)$: The probability that $t$exists in Class C_i

$P(\bar{t}|C_i)$: The probability that $t$is not exists in Class C_i

$P(t|\overline{C_i})$: The probability that $t$exists in all classes except Class C_i

CHI2(Chi Square)

- $CHI^2(t) = \sum_{i=1}^{M} P(C_i)\ CHI2(t, C_i)$
- $CHI^2(t, C_i) = \sum_{r \in \{0,1\}} \sum_{c \in \{0,1\}} \frac{(N_{r,c} - E_{r,c})^2}{E_{r,c}}$

N:Observed frequency, E:Expected frequency

As shown in Table 1, DFS and CHI2 calculate the number of cases based on probabilities. Then, each word is used as a criterion of the selection of qualities by calculating the influence of probability. DFS has a large value when there are lexicon in several clusters and a small value when it is in a single cluster. A smaller value is a valid value, and a CHI2 is a valid value as long as there is no difference in the frequency of clusters.

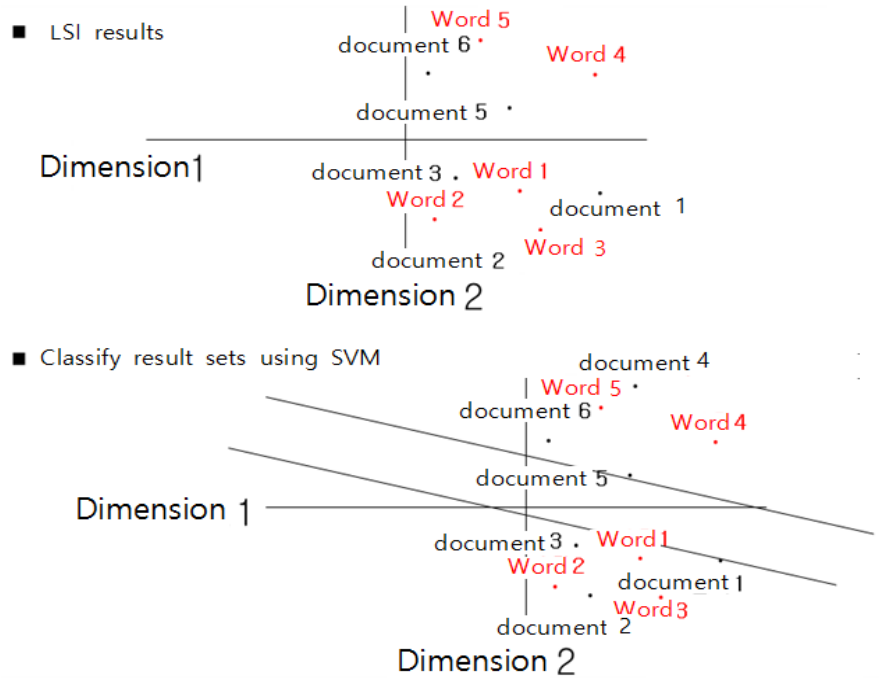

Figure 1. Primary Classification using LSI

GAOLSF proceeds in the following order. As shown in Figure 1, the calculation result is firstly obtained through the LSI, and it is evaluated through DFS and CHI2 in Table 2.

Table 2. DFS Calculation

$$DFS(t) = \sum_{i=1}^{M} \frac{P(C_i|t)}{P(\bar{t}|C_i) + P(t|\bar{C}_i) + 1}$$

$$DFS(term1) = \frac{1}{0+1+1} + \frac{0}{1+0+1} = \frac{1}{2}$$

$$DFS(term2) = \frac{1}{0+1+1} + \frac{0}{1+0+1} = \frac{1}{2}$$

$$DFS(term3) = \frac{1}{0+1+1} + \frac{1}{0+1+1} = 1$$

$$DFS(term4) = \frac{0}{1+0+1} + \frac{1}{0+1+1} = \frac{1}{2}$$

$$DFS(term5) = \frac{0}{1+0+1} + \frac{1}{0+1+1} = \frac{1}{2}$$

As shown in Table 1, the high-value lexicon 3 is removed using the classification result of the DFS, and the classification is performed using the potential semantic index again. Figure 2 shows the result of reclassification.

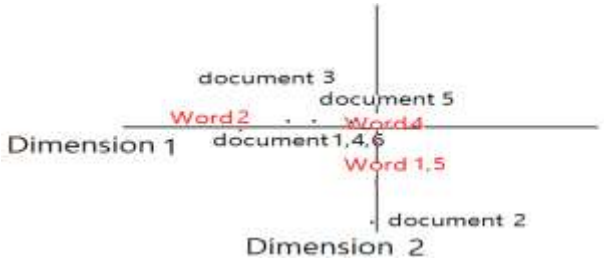

Figure 2. Result of Reclassification

Figure 2 shows the result of reclassification after removing lexicon 3 from its qualities. As shown in Figure 3, it can be seen that the two sets of cosine similarity are completely separated. GAOLSF enhanced clustering through validation of

qualities through genetic algorithms. However, it is difficult to verify selected lexicon because it does not show purpose in resolving ambiguity of lexicon.

2.2. Fuzzy Association Rules and WordNet for Document Clustering

F2IDC is a paper that attempts to document clustering using fuzzy rules and WordNet. The difference from the general clustering method is shown in Figure 3.

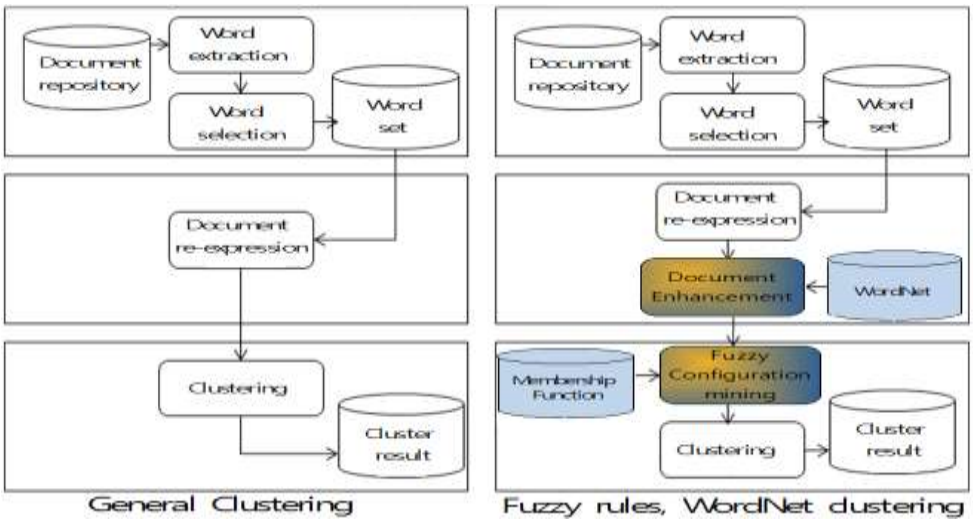

Figure 3. Flowchart of General Clustering and Fuzzy Rules and Word Net Clustering

As shown in Figure 3, F2IDC differs in that it strengthens documents through word-net repository, mining using membership functions, and clustering them.

	Sale	Trade	medical	Health
Document 1	1	0	0	0
Document 2	0	3	0	0
Document 3	1	3	0	0
Document 4	0	0	8	1
Document 5	0	0	11	2

Original matrix

	Sale	Trade	Medical	Health	Marketing	Commerce
Document 1	1	0	0	0	1	1
Document 2	0	3	0	0	0	3
Document 3	1	3	0	0	1	4
Document 4	0	0	8	1	0	0
Document 5	0	0	11	2	0	0

Hyperonym addition matrix

Figure 4. Word Net Reference Conversion

Figure 4 shows the conversion of the original table to the parent word additive matrix by referring to the WordNet relation. As shown in Figure 5, the frequency of each word is calculated as Low, Mid, and High, and converted into a value between 1 and 2 as a fuzzy value.

	Sale			Trade			Medical			Health			Marketing			Commerce		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
Document 1	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	2.0	1.0	1.0
Document 2	0.0	0.0	0.0	1.3	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.3	2.0	1.0
Document 3	2.0	1.0	1.0	1.3	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	1.0	1.0	2.0	1.0
Document 4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	2.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Document 5	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.7	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Total	4.0	2.0	2.0	2.6	4.0	2.0	2.0	2.0	3.0	3.7	3.0	2.0	4.0	2.0	2.0	4.3	5.0	3.0

Figure 5. Apply Fuzzy Rules

As shown in Figure 5, the five original matrix vocabularies are converted into matrices with a total of seven vocabularies by adding marketing and commerce, which are the products of the sale and trade of word-net relations. The fuzzy rules applied in Figure 5 are summarized in Document-Term Matrix (DTM) in Figure 6. The Term-Cluster Matrix can be defined as Figure 7 using the occurrence frequency of words among DTM-organized vocabularies.

	Sale(L)	Trade(M)	Health(L)	Marketing(L)	Commerce(M)
Document 1	2.0	0.0	0.0	2.0	1.0
Document 2	0.0	2.0	0.0	0.0	2.0
Document 3	2.0	2.0	0.0	2.0	2.0
Document 4	0.0	0.0	2.0	0.0	0.0
Document 5	0.0	0.0	1.7	0.0	0.0

Figure 6. Document-Term Matrix(DTM)

The document-term matrix of Figure 6 can be obtained by summarizing the values exceeding the support in the fuzzy rule applied table in Figure 7. The degree of support is a value of the total / document number value of 0.7 or more.

	C(Sale)	C(trade)	C(Health)	C(Marketing)	C(Commerce)
Sale(L)	1.0	0.5	0.0	1.0	1.0
Trade(M)	0.5	1.0	0.0	0.5	1.0
Health(L)	0.0	0.0	1.0	0.0	0.0
Marketing(L)	1.0	0.5	0.0	1.0	1.0
Commerce(M)	0.6	0.8	0.0	0.6	1.0

Figure 7. Term-Cluster Matrix(TCM)

The TCM in Figure 7 summarizes the sum of the concurrent values in the DTM documents in Figure 6 divided by the total value. For example, Sale (L) and C (Sale) have a value of 4/4 by dividing the sum of document 1 and document 3, which is Sale (L), by the sum of columns C (Sale). In addition, Commerce (M) and C (Sale) have a value of 3 and Commerce (M) sum of 5 and 3/5.

Finally, the matrix-multiplication operation of the DTM and the TCM allows the clusters to be distinguished by the Document-Cluster Matrix of Figure 8.

	C(Sale)	C(trade)	C(Health)	C(Marketing)	C(Commerce)
Document 1	4.6	2.8	0.0	4.6	5.0
Document 2	2.2	3.6	0.0	2.2	4.0
Document 3	6.2	5.6	0.0	6.2	8.0
Document 4	0.0	0.0	2.0	0.0	0.0
Document 5	0.0	0.0	1.7	0.0	0.0

Figure 8. Document-Cluster Matrix(DCM)

The Document-Cluster Matrix in Fig 8 is obtained by multiplying the DTM in Figure 6 by the TCM matrix in Figure 7. This indicates that each document belongs to the cluster with the highest value. In conclusion, Document 1, Document 2, and Document 3 belong to the Commerce Cluster, and Document 4 and Document 5 belong to the health cluster. F2IDC was clustering using WordNet. However, there is no interpretation of the relation with many upper and lower words in using WordNet, and there is no verification about it.

3. System Design

3.1. System-wide Flow Chart

The Genetic Algorithm and WordNet based Clustering implemented in this paper are based on the classification using Latent Semantic Indexing, and reinforce the document matrix by referring to WordNet and verify it using Genetic Algorithm. In this section, the overall system overview of the Genetic Algorithm and the WordNet-based Clustering system flow chart is described and briefly describes the partial steps of the system. Figure 9 is a system flow chart.

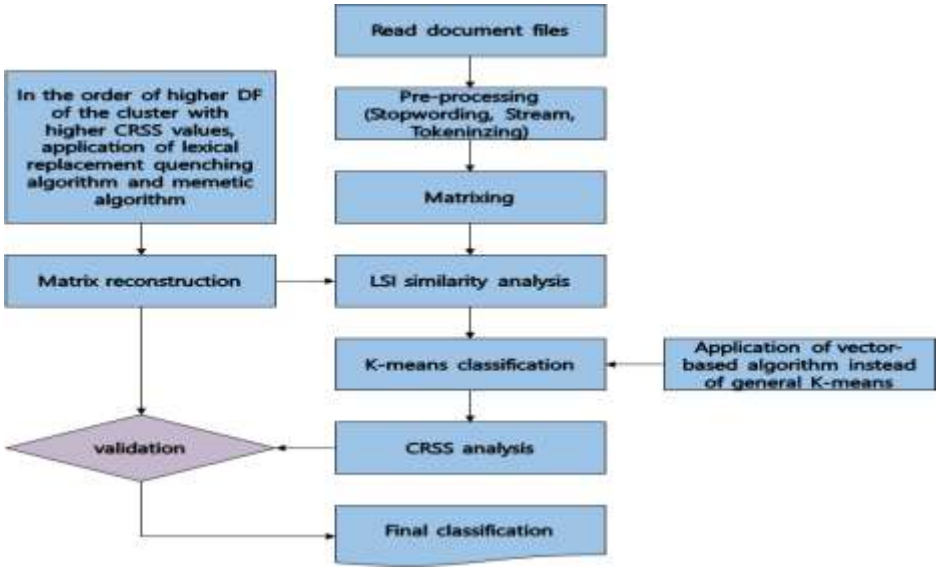

Figure 9. System Flow Chart

As shown in Figure 9, file loading, pre-processing, Matrixization, LSI analysis, K-means classification, CRSS analysis and matrix reconstruction are developed.

File loading is a process of reading a document file to be used for classification while browsing directories and reading WordNet RDF files. The pre-processing process removes the read document (Remove Stopword), root extraction (Stemming) and token extraction (Tokenizing). At this time, not only the general document but also the about attribute and the instance attribute of the WordNet RDF file are processed in the same manner for the same vocabulary. The matrixization operation is a task of counting vocabularies of document data and expressing the weights as matrices according to the formula of TF-IDF (Term Frequency-Inverse Document Frequency). The LSI analysis produces the analysis results by performing Singular Value Decomposition (SVD). Clustering is performed using SVD results in K-means classification. Unlike the conventional K-means clustering, a vector-based method is used. The CRSS analysis is an analytical technique in which the residual sum of squares (RSS) is transformed to an indicator for shifting the center point in K-means clustering. For more details, see Section 2 of this chapter. Based on the results analyzed by CRSS, matrix reconstruction uses WordNet 's rules of Genetic Algorithm to reconstruct the matrix. For more information, see Section 3.

3.2. CRSS Analysis

In this section, we explain the cosine residual sum of squares (CRSS) that is used in the K-means algorithm. Originally K-Means RSS is used as a variant for geometric classification, which is suitable for geometric Euclidean analysis but not for vector models. Therefore, we propose a modified CRSS. CRSS is derived from two formulas of cosine similarity and RSS.

$$similarity = \cos(\Theta) = \frac{A \bullet B}{|A| \, |B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (A_i)^2}} \tag{1}$$

(Equ 1) has a cosine similarity value of -1 when the two individuals A and B are facing the same direction, and when they are orthogonal to each other, when they face the opposite direction of 0. That is, the difference with respect to the direction can be known.

$$\begin{aligned} RSS &= \sum_{1}^{dsc} (C_{center} - d) \\ &= \sum \sqrt{(C_{center1} - d_1)^2 + (C_{center2} - d_2)^2 + \ldots . (C_{centern} - d_n)^2} \end{aligned} \tag{2}$$

(Equ 2) is the equation of the residual sum of squares. Using the Euclidean distance formula, the distance between the document and the center point in the cluster is used to obtain the degree of dispersion in the cluster. The CRSS derived from (Equ 1) and (Equ 2) above are as follows.

$$Cos(s) = \frac{\sum_{i=1}^{n} A_i \bullet B_i}{\sqrt{\sum_{i=1}^{n} A_i^{\,2}} \bullet \sqrt{\sum_{i=1}^{n} B_i^{\,2}}} \tag{3}$$

As shown in (Equ 3), CRSS starts from the cosine similarity. (Equ 3) determines the degree of similarity between two objects A and B. In other words, when the two objects A and B are vectors in the same direction, A and B form an angle of 90 degrees. However, the goal of CRSS is to modify the following equation (Equ 4) by aiming at the sum of the distance differences.

$$\therefore 1 = \frac{\sum_{i=1}^{n} A_i \bullet B_i}{\sqrt{\sum_{i=1}^{n} A_i^{\,2}} \bullet \sqrt{\sum_{i=1}^{n} B_i^{\,2}}} \tag{4}$$

We can express the distance difference between each object by taking the reciprocal of 1 as shown in (Equ 4). If the cosine similarity has a negative value, it should be taken as an absolute value, but this probability is excluded because there is no probability of being the same cluster.

The value of (Equ 4) is the distance between objects, and in a real document cluster, the distance between the center point and the document. Therefore, the object A is represented by the center point C, and the object B is represented by the document D. The value of the sum of the vector difference of the document and the center point in the cluster becomes CRSS, and expressed as (Equ 5).

$$CRSS = \sum_{i=1}^{D_n} (1 - \frac{\sum_{c=1}^{n} C_{ci} \bullet D_w}{\sqrt{\sum_{c=1}^{n} C_i^{\,2}} \bullet \sqrt{\sum_{w=1}^{n} D_w^{\,2}}})) \tag{5}$$

CRSS is finally derived as shown in (Equ 5).

The purpose of RSS in K-means is to move the center of gravity until there is no RSS change in order to find the cluster center. In this system, CRSS verifies cluster analyzed by LSI. The reason why we take this method is that WordNet does not have a probabilistic vocabulary because the vocabulary has no meaning when the same vocabulary has many meanings. Therefore, it is used as a method to verify this in a range as possible through the global search method Genetic Algorithm.

3.3. Genetic Algorithm

This section shows the design of a genetic algorithm that can verify that WordNet is correctly applied. The Genetic Algorithm consists of four steps: Selection, CrossOver, Mutation, and Substitution. In this paper, we propose a new algorithm that uses Genetic Algorithm, Memetic Algorithm, Simulated Annealing. The mimetic technique is a technique that uses local search rather than the global search technique used in the original Genetic Algorithm. The original Genetic Algorithm performs the operation while changing the candidates of a number of solutions, but the quenching technique performs the operation while changing one solution.

The solution is also called Gene (Gene), and the operations that change Gene are the operations in Table 3 above. If you change Gene and change a lot of content at once, you lose the possibility of a global search. Therefore, the DF (Document Frequency) is searched for in the descending order for the global search.

The data structure used as a solution of the Genetic Algorithm is a document-lexical matrix, and the data is represented by an integer two-dimensional matrix. The data of the two-dimensional matrix is the number of vocabularies appearing in the document. Table 3 describes the role of each operation.

Table 3. Genetic Algorithm Operation

Selection	The first document-lexical matrix is selected
CrossOver	If there is one WordNet parent word in the vocabulary
Mutation	If there is more than one WordNet parent word in the vocabulary
Substitution	If CRSS of current value is not equal to or less than CrossOver or Mutation result, exchange value.

The application of the mimetic technique does not change the entire gene, but presupposes a change in the unit cluster. The reason for this is because it is very likely that different bar lexical meaning according to the cluster area. For example, if trade has a vocabulary equivalent to Change and Commerce, the meaning of Change in the case of a sports cluster will be close to the meaning of Commerce in a case of a commercial cluster. Therefore, since the same vocabulary may be written in different meanings depending on the cluster, a mimetic algorithm, which is a local search algorithm, is applied.

The reason for using the quenching technique is as follows. First, dividing the solution into several solutions does not improve performance. Second, systematic limitations require approximately 200 GB of memory for approximately 2000 documents. Even if five solutions are used, the amount of data accumulated in the heap memory exceeds 1TB, so there is no performance advantage.

4. System Implementation

4.1. Implementation Environment

The experimental environment constructed to implement Genetic Algorithm and WordNet based Clustering implemented in this paper is as follows. We used Windows 7 as the operating system and the program used JAVA 1.8.0_60-b27 as the language. The LSI computation was performed using the JAMA (JAVA Matrix Package) 1.0.3 library of the National Institute of Standards and Technology (NIST). WordNet version 2.0 was used, and the document data used in the experiment was NewsGroup 20 (NG20). Direct implementations include abstraction processing, torque estimation, matrix multiplication, and TF-IDF. The hardware used in the experiment was an Intel I-5 CPU, 16GB of RAM, and 12GB of heap memory in Eclipse.

4.2. File Loading and Pre-processing

The file loading and pre-processing module is a process for reading WordNet RDF files and NewsGroup 20 documents and converting them to matrices. First, I used the JAVA DOM Library to read the WordNet RDF file. I split the WordNet RDF file because the DOM (Document Object Model) cannot process large files. NG20 has read over 9,000 documents out of 20,000 documents.

4.3. Matrixization

As a module for performing matrixization, the loading and preprocessing module is a process for reading WordNet RDF files and NewsGroup 20 documents and converting them into matrices. The matrixization module returns the final result in three arrays. The first array is a document array, which stores the document filename as an array. The second array stores the vocabularies in a lexical array. The third array stores the values of the document-matrix in a two-dimensional array

4.4. LSI Similarity Analysis

LSI similarity analysis was performed using Singular Value Decompostion method of JAMA library. Table 4 shows the use of the SVD library.

Table 4. Genetic Algorithm Operation

```
tfidf = tid.tfidf(A);
System.out.println("tfidf");
DecimalFormat formm = new DecimalFormat("0.000");

for(int i=0; i< tfidf.r; i++){
        for(int j=0; j<tfidf.c;j++){
System.out.print(formm.format(tfidf.v[i][j])+" ");
        }
}
double[][] temp;
temp = new double[A.r][A.c];
for(int i=0; i< tfidf.r; i++){
        for(int j=0; j<tfidf.c;j++){
                temp[i][j] = tfidf.v[i][j];
        }
}
Matrix svdf = new Matrix(temp);
Jama.SingularValueDecomposition SVD = new SingularValueDecomposition(svdf);
u = SVD.getU();
s = SVD.getS();
vt = SVD.getV();
vt = vt.transpose();
matrix ua= new matrix(u.getRowDimension(), u.getColumnDimension());
matrix sa= new matrix(s.getRowDimension(), s.getColumnDimension());
matrix vta= new matrix(vt.getRowDimension(), vt.getColumnDimension());
matrix sd = new matrix(2, 2);
matrix ud = new matrix(u.getRowDimension(), 2);
matrix vtd = new matrix(2, vt.getColumnDimension());
```

Table 4 converts the matrix obtained by the self-implemented tf-idf method into the matrix of u, s, vt respectively by SVD objects obtained by SingularValueDecompostion of JAMA library.

4.5. K-Means Clustering

LSI similarity analysis is first calculated by U × Σ. Table 5 is the cosine similarity of the U × Σ matrix. Clustering is performed using the results obtained in Table 5.

Table 5. U × ∑ Cosine Similarity

```
public static matrix pcsim(matrix a){
int msize;
if(a.r > a.c){
msize = a.r;
}else{
msize = a.c;
}
matrix result = new matrix(msize, msize);
for(int i=0;i<a.c;i++){
for(int j=0; j<a.c; j++){
if(i == j){
result.v[i][j] = 1;
}else if(i > j){
                        double child = 0;
                        double mother1 = 0;
```



```
                        double mother2 = 0;
                        for(int k=0; k<a.r; k++){
                         child += a.v[k][i] * a.v[k][j];
                         mother1 += a.v[k][i] * a.v[k][i];
                         mother2 += a.v[k][j] * a.v[k][j];
                         double mother;
                         mother = Math.sqrt(mother1) * Math.sqrt(mother2);
                         result.v[i][j] = Double.parseDouble(String.format("%.3f", child / mother));
                        }
            }else{
                        result.v[i][j] = 0;
            }
}
}
return result;
}
```

4.6. CRSS Operation

CRSS operations are described in Section 3 and used for validation. Are obtained by the following Table 6.

Table 6. U × ∑ CRSS Implement

```
public double calCRSS(double[] centerV, double[][] matUSig){
double result=0;
double centerMom=0;
double docMom=0;
double child = 0;
/*
            Since the Sigma matrix is always a square matrix, the number of center [] and the number of
matUSig [j] [] are always matched.
*/
for(int j=0; j < matUSig.length;j++){
            for(int i = 0; i < centerV.length; i++){
             centerMom += Math.sqrt(Math.pow(centerV[i], 2));
            }
            for(int i = 0; i < centerV.length; i++){
             docMom += Math.sqrt(Math.pow(matUSig[j][i], 2));
            }
            for(int i = 0; i < centerV.length; i++){
child += centerV[i] * matUSig[j][i];
            }
            result += 1-(child/centerMom*docMom);
}
return result;
}
```

4.7. Matrix Reconstruction

Matrix reconstruction operations are handled using WordNet and the Genetic Algorithm. Table 7 shows the matrix reconstruction module. In Table 7, matrix reconfiguration changes the matrix by referring to WordNet.

Table 7. Matrix Reconstruction

```
public static int[][] remat(String[] wordnet, String[] docs, String[] words, int[][] mat){
for(int k=0; k<wordnet.length-1;k++){
if(!wordnet[k].split(";")[0].equals(wordnet[k].split(";")[0])
            &&words[k].equals(wordnet[k].split(";")[0])){
            for(int j=0;j<docs.length;j++){
            int temp = mat[k][j];
                        mat[k][j] = 0;
                        for(int i=0;i<wordnet.length;i++){
```

```
                        mat[k][i] = temp;
                       }
              }
             }
}
return mat;
}
```

4.8. Cluster Analyzer

The cluster analyzer projects the U × Σ matrix onto the multidimensional space and analyzes it using the cosine similarity between the objects and the CRSS. Table 8 shows the contents of the cluster analyzer. Table 8 shows the implementation of the cluster analyzer.

Table 8. Cluster Analysis

```
public static double[][] ClusterAnalizer(int n, double[][] cossim){
Random r = new Random();
double[][] clust = new int[n][cossim.length];
for(int i=0; i<n; i++){
            int cluster_n = r.nextInt(cossim.length);
            clust[i][0] = cluster_n;
for(int j=1; j<cossim[0].length;j++){
             if(cossim[i][j]>0.7){
                        clust[i][j] = cossim[cluster_n][j];
             }
}
            clust = mvclust(cossim, clust);
}
return clust;
}

public static double[][] mvclust(double[][] cossim, double[][] clust){
for(int i = 0; i<clust.length-1;i++){
            CRSS.calCRSS(clust[i], clust);
}
return clust;
}
```

5. Performance Evaluation

5.1. Performance Evaluation Environment

The PC used for performance evaluation was Intel i-5 2450K, 16GB RAM, OS was Windows7, and Eclipse console was used. The result was output to a text file. The document used in the experiment is NewsGroup20. NewsGroup20 is classified into 20 items. It consists of 20,000 documents, 1000 documents per item. In this paper, we randomly extracted 500 items from 20 items and experimented with 10,000 documents. The total number of vocabulary extracted by pre-processing is 29332, and about 90,000 words are extracted by WordNet.

The experiment is based on extracted results up to 20 times by inserting 10,000 documents into GWLSI in bulk and compared with the cluster classified based on the answers provided by NewsGroup20. The evaluation items are Precision, Recall, Accuracy, F1-Measure, and CRSS.

5.2. Performance Evaluation Items

Precision, Recall, Accuracy, and F1-Measure used for performance evaluation are shown in Table 9.

Table 9. Performance Evaluation Item

		Experiment result	
		True	False
Actual answer	True	A True Positive	B False Posivite
	False	C True Negative	D False Negative

As shown in Table 9, the experimental results are compared with the actual answers, and they are differentiated through Table 10. Zone A is consistent with the answers provided by Cluster and NewGroup 20, which are classified as GWLSI, and Zone B is not present in the Cluster of GWLSI but exists in the answer of NewsGroup 20. Zone C is in the Cluster of GWLSI but not in the answer of NewGroup20, and Zone D is not in the answer of GWLSI and NewsGroup20.

Table 10. Performance Evaluation Item Define

Precision	A/(A+C)
Recall	A/(A+B)
Accuracy	(A+D)/(A+B+C+D)
F1-Measure	2*P*R/(P+R)

As you can see in Table 10, Precision is asking how well the actual answer came out, and it is the concordance rate of the item of NewsGroup 20 and the answer of GWLSI. Recall asks how precisely the results of the experiment are coming from and assesses how exactly the GWLSI classification results match the actual answer. Accuracy is an item that checks how precisely the whole item is coming from, and it shows the exact clustered accuracy in the entire document. F1-Measure is the harmonic mean of Precision and Recall. CRSS is the degree of dispersion inside the cluster, which means cluster density. The higher the density, the more advantageous it can be for machine learning. The higher the density of clusters, the higher the degree of integration.

5.3. Performance Evaluation

The performance was evaluated using the results of LSI and GWLSI 20 times. The documents used in the experiment were processed by using NewGroup20 documents with 10,000 documents, 30,000 vocabularies, and WordNet's 90,000 relationships. The performance evaluation items are Precision, Recall, Accuracy, F1-Measure, and CRSS. Precision's results, which show the accuracy of the experimental results, are shown in Figure 10.

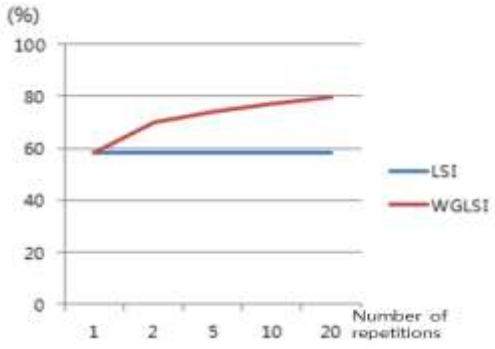

Figure 10. Precision Result

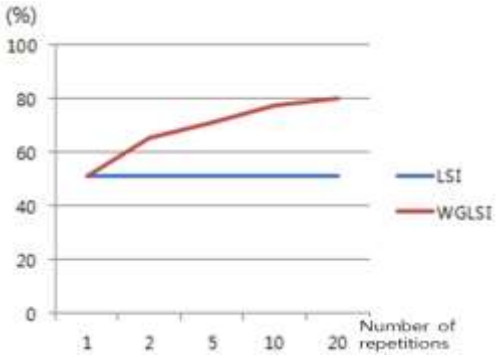

Figure 21. Recall Result

As shown in Figure 10, it shows the same performance in the first round but it rises sharply in the second round, and then the performance improves at a satisfactory speed. 20 times more than 20% of the performance improvement. The result of Recall, which shows the recall rate of the entire document, is shown in Figure 11. As shown in Figure 11, it shows the same performance in the first round but it rises sharply in the second round, and then the performance improves at a satisfactory speed. It shows about 30% performance improvement from 20 times. The results of Accuracy measuring the correct classification in the whole document are shown in Figure 12.

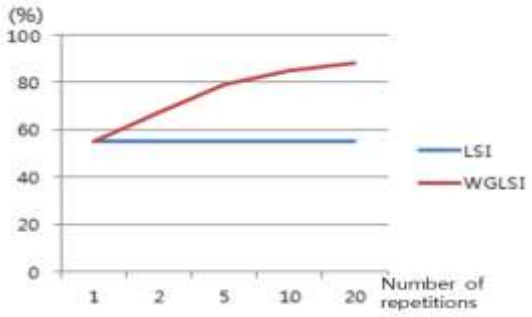

Figure 32. Accuracy Result

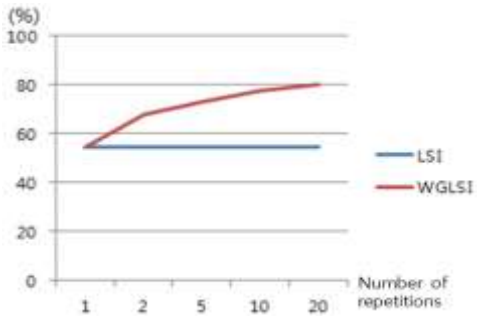

Figure 43. F1-Measure Result

As shown in Figure 12, the same performance is shown in the first round but it rises sharply in the second round and then the performance improves at a satisfactory speed. It shows about 35% performance improvement from 20th. The result of F1-Measure, which is the harmonic mean value of Precision and Recall, is shown in Figure 13. As shown in Figure 13, it shows the same performance in the first round but sharply rises in the second round, and then the performance improves at a satisfactory speed. It shows a performance improvement of about 25% from 20th. CRSS is divided by the number of documents in the cluster because there is a possibility that it may become infinitely large as the number of documents increases, and the average result of clusters is reported. The results are shown in Figure 14.

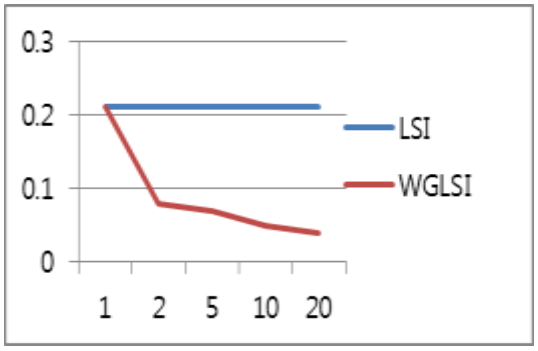

Figure 54. CRSS Result

As shown in Figure 14, the same performance is shown in the first round, but it rises sharply in the second round and then the performance improves at a satisfactory speed. In the 20th period, the value dropped by about 17%, indicating that the documents are arranged close to each other. Thus, you can see that the documents are fragmented on the subject.

6. Conclusion

In this paper, we did not use existing geometric methods to preserve the storage of potential semantic index-based documents using the Genetic Algorithm and WordNet-based Clustering. Instead, CRSS, a method of vector method analysis, and a genetic algorithm for verifying and applying it are described. We also proposed a new clustering method that can store data through a cluster analyzer.

Genetic Algorithm and WordNet-based Clustering developed in this paper can be applied not only to storage but also to translation, autocomplete, document correction, recommendation, emotional analysis. However, since the system of this paper is made from the viewpoint of storage, it is necessary to study the modification of the algorithm and the index in order to apply it to other directions. Future work will require research on indexing techniques for speeding up searches and information retrieval such as search, ranking, and translation.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2015-0-00376, IoT-based CPS platform technology for the integration of virtual-real manufacturing facility)

This paper is a revised and expanded version of a paper entitled [Bigdata based Network Traffic Feature Extraction] presented at [2017 the 14th international workshop series, daejeon university, korea, 2017.12.21~2017.12.23]

References

[1] C. L. Chen, F. S. Tseng and T. Liang, “An Integration of Fuzzy Association Rules and WordNet for Document Clustering”, Journal of Knowledge and Information Systems, vol. 28, no. 3, **(2011)**, pp. 687-708.
[2] M. B. David, Y. N. Andrew and I. J. Michel, “Latent Dirichlet Allocation”, Journal of Machine Learning Research, vol. 3, **(2003)**, pp. 993-1022.
[3] C. Deng, H. Xiaofei and H. Jiwei, “Document Clustering Using Locality Preserving Indexing”, Journal of IEEE Transaction on Knowledge and Engineering, vol. 17, no. 12, **(2005)**, pp. 1624-1637.
[4] T. K. Landeuer, P. W. Foltz and D. Laham, “Introduction to Latent Semantic Analysis”, Journal of Discourse Processes, vol. 25, no. 2-3, **(1998)**, pp. 259-284.
[5] A. G. Miller, “Word Net : A Lexical Database for English”, Journal of Communication of the ACM, vol. 38, no. 11, **(1995)**, pp. 39-41.

[6] Z. Taiping, Y. T. Yuan, F. Bin and X. Yong, "Document Clustering in Correlation Similarity Measure Space", Journal of IEEE Transaction on Knowledge and Data Engineering, vol. 24, no. 6, **(2012)**, pp. 391-407.
[7] G. Teng, Y. Xia, E. Camria, P. Jin and T. F. Zheng, "Document representation with statistical word senses in cross-lingual document clustering", Journal of Pattern Recognition and Artificial Intelligence, vol. 29, no. 2, **(2015)**, 1559003(26pages).
[8] A. K. Uysal and S. Gunal, "Text Classification Using Genetic Algorithm Oriented Latent Semantic Features", Journal of Expert Systems with Applications, vol. 41, no. 13, **(2014)**, pp. 5938-5947.
[9] T. Wei, Y. Lu, H. Chang, Q. Zhou and X. Bao, "A Semantic Approach for Text Clustering using WordNet and Lexical Chains", Journal of Expert Systems with Applications, vol. 42, no. 4, **(2015)**, pp. 2264-2275.

Authors

Jeong-Joon Kim, received his BS and MS in Computer Science at Konkuk University in 2003 and 2005, respectively. In 2010, he received his PhD in at Konkuk University. He is currently a professor at the department of Computer Science at Korea Polytechnic University. His research interests include Database Systems, BigData, Semantic Web, Geographic Information Systems (GIS) and Ubiquitous Sensor Network (USN), etc. e-mail: jjkim@kpu.ac.kr

Yong-soo Lee, received his MS in Computer Science at Konkuk University in 1989. In 2015, he received his PhD in Information & Control Engineering at Kwangwoon University. He is currently a professor at the Department of Computer Information at Yeoju Institute of Technology. He is the Member of the Korea Institute of Internet, Broadcasting & Communication (IIBC). His research interests include Database Systems, Data Mining, BigData, Wireless Sensor Networks and Ubiquitous Sensor Network (USN), etc. e-mail: diclee@yit.ac.kr

Jin-Yong Moon, received his BS in Computer Science at Suwon University in 1996. He received his MS in Computer Science at Konkuk University in 1998. Then he received Ph.D. degree from Suwon University in 2001. He is currently a professor in the department of Visual Broadcasting Media at Gangdong College. His research interests include Database Systems, Mobile Systems, Geographic Information Systems (GIS) and Multimedia Systems, etc. e-mail: jmoon37@gmail.com

Jeong-Min Park, received his Ph.D. and M.S. degrees in Department of Computer Engineering from Sungkyunkwan University, Korea, in 2009 and 2005, respectively, and his B.S. degree in Computer Engineering from Korea Polytechnic University, in 2003. Currently, He is currently an assistance professor of the Department of Computer Engineering at Korea Polytechnic University, Korea. From 2012 to 2014, he was a senior member of engineering staff, in ETRI, Korea. His research interests include Cyber-Physical System (CPS), Autonomic Computing and Software Engineering. e-mail: jmpark@kpu.ac.kr

