Document Representation and Clustering with WordNet Based Similarity Rough Set Model

Nguyen Chi Thanh and Koichi Yamada

Department of Management and Information System Science, Nagaoka University of Technology, Nagaoka-shi, 940-2188 Japan

Abstract

Most studies on document clustering till date use Vector Space Model (VSM) to represent documents in the document space, where documents are denoted by a vector in a word vector space. The standard VSM does not take into account the semantic relatedness between terms. Thus, terms with some semantic similarity are dealt with in the same way as terms with no semantic relatedness. Since this unconcern about semantics reduces the quality of clustering results, many studies have proposed various approaches to introduce knowledge of semantic relatedness into VSM model. Those approaches give better results than the standard VSM. However they still have their own issues. We propose a new approach as a combination of two approaches, one of which uses Rough Sets theory and co-occurrence of terms, and the other uses WordNet knowledge to solve these issues. Experiments for its evaluation show advantage of the proposed approach over the others.

Keywords: document clustering, document representation, rough sets, text mining.

1. Introduction

Document clustering is an important text mining technique to generate useful information from text collections such as news articles, research papers, books, digital libraries, e-mail messages, and web pages. Text-based document clustering attempts to group documents into clusters where each cluster might represent a topic that is different from topics of the other clusters.

Document clustering algorithms are divided into two categories in general: partitional clustering and hierarchical clustering. Partitional clustering divides a document collection into groups in a single level, while hierarchical clustering creates a tree structure of documents. There are various document clustering methods proposed in recent years, including hierarchical clustering algorithms using results from a k-way partitional clustering solution [1], spherical k-means [2], bisecting k-means [3], frequent term meaning sequences based method [4], k-means with Harmony Search Optimization [5].

Vector space model is a popular model for document representation in document clustering including the above methods. Documents are represented by vectors of weights, where each weight in a vector denotes importance of a term in the document. In the standard VSM, however, semantic relations between terms are not taken into account. Two terms with a close semantic relation and two other terms with no semantic relation are both treated in the same way. This unconcern about semantics could reduce quality of the clustering result.

There are some approaches proposed to deal with this problem. Tolerance Rough Set model (TRSM) [6] and Similarity Rough Set Model (SRSM) [7] extended the vector space model using Rough Sets theory and co-occurrence of terms. TRSM and SRSM have been successfully applied to document clustering. However, the results showed that SRSM had better performance than TRSM and some other conventional methods [7].

There are other approaches that employ WordNet based semantic similarity to enhance the performance of document clustering [8, 9]. They modified the VSM model by readjusting term weights in the document vectors based on its relationships with other terms co-occurring in the document.

SRSM and WordNet based methods performed better results than the standard VSM. However, they still have their own issues as discussed later. We propose a new method by combining their strength and reducing their weakness. The new method uses both Rough Sets theory and WordNet based semantic similarity to define a new representation model of documents. Experimental results show that it gives better clustering results than the other methods discussed in the paper.

The paper is organized by six sections. In Section 2 and Section 3 we discuss SRSM and WordNet semantic similarity based methods, respectively. Section 4 describes our proposed method. Section 5 presents the results of our experiments on document collections. Finally, Section 6 concludes with a summary and discussion about future research.

2. Similarity rough set model

Similarity Rough Set Model is a mathematical model extended from Pawlak's Rough Set model [10] using similarity relation instead of equivalence relation [7]. It is also an expansion from Tolerance Rough Set Model [6] with a tolerance relation.

Equivalence, tolerance and similarity relations are binary relations that could be used to represent relations between terms in document clustering. An equivalence relation must satisfy reflexive, symmetric and transitive properties, while a tolerance relation does not have to satisfy transitive one. A similarity relation must be reflexive, but not required to be symmetric and transitive [11, 12].

TRSM based on a tolerance relation was successfully applied to information retrieval and document clustering in [6, 13, 14]. Recently, SRSM based on a similarity relation was proposed and applied to document clustering by authors of this paper [7]. It showed that SRSM produces better results than TRSM both in quality and robustness, where co-occurrence of terms was used to obtain tolerance and similarity relations, respectively.

SRSM could be defined as follows: Let the pair $apr = (U, R)$ be an approximation space, where U is the universe, and $R \subset U \times U$ is a similarity relation on U.

$r(x)$: $U \rightarrow 2^U$ is an uncertainty function which corresponds to the similarity relation R understood as $yRx \Leftrightarrow y \in r(x)$, which might represent that y is similar to x. $r(x)$ is a similarity class of all objects that are considered to have similar information to x. The function $r(x)$ satisfies reflexive property: $x \in r(x)$, however it is not necessary symmetric and transitive.

Given an arbitrary set $X \subset U$, X can be characterized by a pair of lower and upper approximations as follows:

$$\underline{apr}(X) = \left\{x \in U \mid r^{-1}(x) \subset X\right\}, \quad (1)$$

$$\overline{apr}(X) = \bigcup_{x \in X} r(x), \quad (2)$$

where $r^{-1}(x)$ denotes the inverse relation of R, which is the class of referent objects to which x is similar:

$$r^{-1}(x) = \left\{y \in U \mid xRy\right\} \quad (3)$$

We proposed a new model of document representation for document clustering using the above generalized rough set theory – Similarity Rough Set Model [7]. The new model is defined as follows.

The universe U of the approximation space (U, R) is the set of all terms T used in the document vectors. The binary relation R is defined by

$$t_j R t_i \Leftrightarrow f_D(t_i, t_j) \geq \alpha . f_D(t_i), \quad (4)$$

where $f_D(t_i, t_j)$ is the number of documents in the document set D in which term t_i and t_j co-occur, $f_D(t_i)$ is the number of documents in D in which term t_i occurs and α is a parameter ($0 < \alpha < 1$). The relation R defined above is a similarity relation that satisfies only reflexivity.

An uncertainty function $I_\alpha(t_i)$ corresponding to the similarity relation is defined as

$$I_\alpha(t_i) = \{t_j \in U \mid t_j R t_i\}, \quad (5)$$

where $I_\alpha(t_i)$ is a set of all terms similar to t_i.

The lower and upper approximation of any subset $X \subset T$ based on this model can be obtained using equations (1) and (2), where U and r are replaced by T and I_α, respectively.

In this case, $I_\alpha^{-1}(t_i)$ is the set of terms to which t_i is similar, and is defined as

$$I_\alpha^{-1}(t_i) = \left\{t_j \mid f_D(t_i, t_j) \geq \alpha . f_D(t_j)\right\} \quad (6)$$

In the document clustering with SRSM (referred to as SRSM later, while the ordinary approach is referred to as VSM), we applied spherical k-means algorithm [2] to term vectors that consists of terms in upper approximations of ordinary document vectors (term sets). The usage of upper approximation could give us better clustering results, because two documents become similar to each other, if one contains many terms similar (in the sense of eq. (4)) to terms in the other even if the two documents do not have many common terms. Since there are many synonyms in natural language in general and people use different terms to represent a certain thing, the upper approximation would give a positive effect on document clustering.

There would be another advantage of using the upper approximation. The number of terms in a document is usually relatively small in comparison with the number of terms in a corpus. Therefore, the document vectors are usually high dimensional and sparse. Hence, document similarity measurements often yield zero values, which can lead to the poor clustering results. Since the proposed approach puts additional terms into document vectors without increasing the dimension, the unwelcome tendency might be mitigated to some extent.

We use tf×idf weighting scheme to calculate the weights of terms in upper approximations of the document vectors. The term weighting method is extended to define weights of terms that are not contained in documents but in the upper approximations. It ensures that such terms have a weight smaller than the weight of any other term in the document. The weight a_{ij} of term t_i in the upper approximation of document d_j is then defined as follows.

$$a_{ij} = \begin{cases} f_{ij} \times \log\left(\frac{N}{f_{D(t_i)}}\right) & \text{if } t_i \in d_j \\ \min_{t_h \in d_j} w_{hj} \times \frac{\log\left(\frac{N}{f_{D(t_i)}}\right)}{1+\log\left(\frac{N}{f_{D(t_i)}}\right)} & \text{if } t_i \in \overline{apr}(d_j) \setminus d_j \\ 0 & \text{if } t_i \notin \overline{apr}(d_j) \end{cases} \quad (7)$$

where f_{ij} is the frequency of term i in document j, N is number of documents, d_j is a set of terms appearing in document j, t_h is the term with the smallest weight in the document j and w_{hj} is the original weight of term t_h in the document j. Then normalization is applied to the upper approximations of document vectors. The cosine similarity measure is used to calculate the similarity between two vectors.

The algorithm is described as follows [7]:

1. Preprocessing (word stemming, stopwords removal).
2. Create document vectors.
   - 2.a. Obtain sets of terms appearing in documents.
   - 2.b. Create document vectors using tf×idf.
   - 2.b. Generate similarity classes of terms based on their co-occurrences.
   - 2.c. Create vectors of upper approximations of documents using equation (7) and then the vectors are normalized.
3. Apply the clustering algorithm
   - 3.a. Start with a random partitioning of the vectors of upper approximations of documents, namely $C^{(0)} = \{C_1^{(0)}, C_2^{(0)}, ..., C_k^{(0)}\}$. Let $c_1^{(0)}, c_2^{(0)}, ..., c_k^{(0)}$ denote the centroids of the given partitioning with the index of iteration $t = 0$.
   - 3.b. For each document vector x_i, $1 \le i \le N$, find the centroid closest in cosine similarity to its upper approximation $\overline{apr}(x_i)$. Then, compute the new partitioning $C^{(t+1)}$ based on the old centroids $c_1^{(t)}, c_2^{(t)}, ..., c_k^{(t)}$:
     $C_j^{(t+1)}$ is the set of all document vectors whose upper approximations are closest to the centroid $c_j^{(t)}$. If the upper approximation of a document is closest to more than one centroid, then it is randomly assigned to one of the clusters.
   - 3.c. Compute the new centroids:

$$s_j = \sum_i \overline{apr}(x_i), \; c_j^{(t+1)} = \frac{s_j}{\|s_j\|}, \; 1 \le j \le k,$$

     where $c_j^{(t+1)}$ denotes the centroid or the mean of the upper approximations of documents in cluster $C_j^{(t+1)}$.
   - 3.d. If some "stopping criterion" is met, then set $C_j^* = C_j^{(t+1)}$ and set $c_j^* = c_j^{(t+1)}$ for $1 \le j \le k$, and exit. Otherwise, increment t by 1, and go to step 3.b above.

In our implementation, the iteration stops when the centroids of the generated clusters are identical to those generated in the previous iteration.

In SRSM, we used co-occurrence of terms to calculate the semantic relation between terms. The usage of co-occurrence gives us a merit that lets us define similarity relations automatically without any knowledge base. However, it might also have a weakness that in some cases co-occurrence of terms does not necessarily mean they have a similar meaning. In the case, terms that do not appear in a document nor similar to any term in the document may be contained in the upper approximation.

3. WordNet semantic similarity based model

WordNet is an electronic lexical database of English, available to researchers in computational linguistics and natural language processing [15]. WordNet was developed and is being maintained by the Cognitive Science Laboratory of Princeton University. In WordNet, a concept represents a meaning of a term. Terms which have the same concept are grouped in a synset. Each synset has its definition (gloss) and links with other synsets higher or lower in the hierarchy by different types of semantic relations.

There are different methods to compute semantic similarity of terms using WordNet, which can be divided into four categories: path based, information content based, gloss based and vector based methods. Path based methods use length of the path between concept nodes to calculate the similarity relatedness [16, 17]. Information content based methods [18, 19] measure the relatedness of the two concepts using the information content of the most specific shared parent. In gloss based methods [20, 21], glosses of concepts are used to determine the relatedness of concepts. In vector based methods [22, 23], the relatedness between terms are computed using concept vectors derived from glosses.

Recently, some studies used WordNet-based semantic similarity to enhance performance of document clustering [8, 9]. They modified the VSM model by readjusting weights of terms in the documents. The basic idea is that a term is considered more important if other terms semantically related to it appear in the same document. They increase weight values of such terms with the following equation:

$$\tilde{w}_{i_1 j} = w_{i_1 j} + \sum_{\substack{t_{i_2} \in d_j \\ i_2 \neq i_1}} sim(t_{i_1}, t_{i_2}) w_{i_2 j} \quad (8)$$

where $w_{i_1 j}$ is the original weights of term t_{i_1} in document d_j, $sim(t_{i_1}, t_{i_2})$ is the semantic similarity between the two terms calculated using a WordNet based measure. They proposed improved VSM model based on this idea and showed that the clustering performance based on the new model was better than that based on the VSM.

The advantage of this approach is the high reliability of similarity given by the WordNet. The basic idea behind eq. (8) also seems adequate. A possible weak point might come from the general property of WordNet. Since it is a general dictionary, it might not work for documents in a specific field. Another is that it utilizes the knowledge of similarity only to adjust the importance of terms in a document. It does not let us find similarity between two documents where one contains many terms similar to ones in the other but the two do not have many common terms.

4. WordNet based similarity rough set model for document clustering

In document clustering, the effect of semantic similarity between terms is large, and must be taken into account to enhance the performance of VSM. In SRSM, the semantic relation between terms is calculated using co-occurrence of terms. However, there seem cases when terms have high co-occurrence but have low semantic similarity. WordNet-based approaches measure the relatedness of terms using the lexical database. Based on the ontology structure of terms or definitions of terms in WordNet, we can compute scores of semantic relatedness. However, as a general dictionary, WordNet does not cover all terms and term meanings in every specific subject. Moreover, in different fields, the semantic relation of terms may be different. Our idea is to exploit both approaches to get better clustering results.

In SRSM, we defined the similarity class of terms using the relation R given by eq. (4). Here, we propose a new relation that integrates WordNet knowledge to eliminate terms having no similar meaning but a high frequency of co-occurrence.

$$t_j R t_i \Leftrightarrow f_D(t_i, t_j) \geq \alpha . f_D(t_i) \wedge ((t_i \text{ not in WordNet}) \vee (t_j \text{ not in WordNet}) \vee \text{sim}(t_i, t_j) > \theta)), \quad (9)$$

where θ is a threshold value.

The relation defined by Eq. (9) is a similarity relation, because it is reflexive, non-symmetric and non-transitive. The basic idea is that term t_j is similar to t_i when t_j is similar to t_i from the viewpoint of co-occurrence and they are also similar in the semantics of WordNet. If t_i or t_j is not in WordNet, we use only the co-occurrence similarity.

Then we can define a new representation model based on this relation in the similar way to the one in section 2.

Let the pair $apr = (U, R)$ be an approximation space, where U is the set of all index terms T in the same way as SRSM, and $R \subset U \times U$ is a similarity relation on U.

$r(x)$: $U \rightarrow 2^U$ is an uncertainty function which corresponds to the relation R understood as $yRx \Leftrightarrow y \in r(x)$, which might represent that y is similar to x. $r(x)$ is a similarity class of all objects that are considered to have similar information to x. The function $r(x)$ satisfies reflexive property: $x \in r(x)$, however it is not necessary symmetric and transitive.

Given an arbitrary set $X \subset U$, X can be characterized by a pair of lower and upper approximations as equations (1) and (2).

The binary relation R is a relation that corresponds to an uncertainty function defined by eq. (9). That is,

$$I_{\alpha\theta}(t_i) = \{t_j \in U \mid t_j R t_i\}. \quad (10)$$

R is a similarity relation because it only satisfies the properties of reflexivity.

In SRSM, we assigned weights to terms that do not occur in the document but belong to similarity classes of terms in the document, and do not change the weight values of terms in the document. In the new method we improve the SRSM by readjusting weight values of terms based on the idea of WordNet based methods.

The weight a_{ij} of term t_i in the upper approximation of document d_j is then defined as follows.

$$a_{ij} = \begin{cases} f_{ij} \times \log\left(\frac{N}{f_{D(t_i)}}\right) + \sum_{\substack{t_k \in d_j \\ k \neq i}} sim(t_i, t_k) a_{kj} & \text{if } t_i \in d_j \\ \min_{t_h \in d_j} a_{hj} \times \frac{\log\left(\frac{N}{f_{D(t_i)}}\right)}{1 + \log\left(\frac{N}{f_{D(t_i)}}\right)} & \text{if } t_i \in \overline{apr}(d_j) \setminus d_j \\ 0 & \text{if } t_i \notin \overline{apr}(d_j) \end{cases} \quad (11)$$

The new proposed approach could be regarded as a combination of SRSM and WSSM (WordNet Semantic Similarity based Model) which incorporate the advantages of both the models. WSSM is completely included in the proposed approach because weights of terms in a document are adjusted using eq. (8). In addition, it is an improved version of SRSM, because it calculates the upper approximation of the term set of a document and uses it as the document vector. The improvements are the similarity relation (eq. (9)) used to calculate the upper approximation, and eq. (11) to readjust the weights of terms that are contained in the document.

5. Experimental results

In the experiments, we use two test collections to evaluate the proposed approach in comparison with SRSM, WSSM and methods in CLUTO toolkit [24]. The algorithms provided in CLUTO toolkit are based on the partitional, agglomerative, and graph-partitioning paradigms. They are denoted as *rb*, *rbr*, *direct*, *agglo*, *graph*, *bagglo*. The *rb* is a repeated bisecting approach. The *rbr* is the same as the repeated bisecting method except that at the end the overall solution is globally optimized. The *direct* is a partition method which uses an iterative refinement algorithm to optimize a global clustering criterion function. The *agglo* is an agglomerative clustering algorithm. The *graph* uses a nearest-neighbor graph to model documents, and then divides the graph into *k* clusters using a min-cut graph partitioning algorithm. In the *bagglo*, agglomeration process is used to cluster documents after the document collection is split into $\sqrt{N}$ clusters using the *rb* method.

The first test collection is a classic data set obtained by combining CACM, CISI, CRANFIELD, and MEDLINE abstracts which is available from [25]. The dataset includes abstracts of papers in different fields. CACM contains 3204 abstracts from Communications of ACM, CISI contains 1460 abstracts of information science papers, CRANDFIELD contains 1400 abstract of aeronautical papers, MEDLINE contains 1033 abstracts of medicine papers. The clustering algorithms are supposed to cluster the dataset containing 7097 abstracts into four groups.

After preprocessing (stemming, stop-words elimination, and high frequency word pruning), we have 13177 terms in the document collection. With the 13177 terms we created 7097 document vectors using tf×idf weighting scheme, each document vector has 13177 dimensions.

We evaluate clustering results obtained by each algorithm with three commonly-used measures: entropy, *F* measure and mutual information [3, 26]. There are different clustering quality measures rendering different results. However, if a method performs better than the others on many of these measures then we could say that the method is better than the others.

Entropy, *F* measure and mutual information measures are external quality measures which evaluate the clustering results by comparing the clusters produced by the algorithm to the known classes of documents. With the entropy measure method, the clustering quality is better if the entropy is smaller. While with *F* measure and mutual information method, the higher the evaluated values are the better clustering result is.

We run the experiments with the proposed method, SRSM and WSSM. We also run the test collection with the CLUTO toolkit.

The WordNet-based similarity measure used in the experiment is the Wu and Palmer measure [17], which is a path-based method. It computes the relatedness of two concepts using the lowest common subsumer of two concepts *lcs*(c_1, c_2) which is the first shared concept on the paths from the concepts to the root concept of the ontology hierarchy.

$$sim(c_1, c_2) = \frac{2 \times depth(lcs)}{l(c_1, lcs) + l(c_2, lcs) + 2 \times depth(lcs)} \quad (12)$$

where $l(c_1, lcs)$ is the length of the path between the two nodes and *depth*(*lcs*) is the number of nodes on the path from *lcs* to root.

We ran the experiment using the proposed method with the value of threshold $\theta = 0.3$.

Table 1 shows the evaluation of clustering results from CLUTO toolkit's algorithms, Table 2 shows the evaluation of clustering results of SRSM and the newly proposed method with different values of parameter α. The best evaluation in each quality measure is shaded in Table 2. As for WSSM, the evaluation of the clustering result was 0.363, 0.332, 0.894 for entropy, mutual information and *F* measure respectively. As seen in these results, the best case is the clustering by the proposed approach with $\alpha = 0.55$ in all three evaluation measures.

With SRSM, co-occurrence of terms is used to determine the similarity classes of terms. In the proposed method, we use both co-occurrence of terms and WordNet based semantic similarity. The new approach, as suggested by the figures in the column of "size of similarity classes" in Table 2, can remove irrelevant terms from similarity classes. For example, with the SRSM implementation in our experiment, similarity class of "photon" contains "integration" and "algebra", which have low semantic relatedness with "photon" itself. With the new method, "integration" and "algebra" are removed from the similarity class. Another example would be the term "program", which for SRSM is in the similarity class of "glossary", while for the new approach it is not the case. The removal of irrelevant terms improves the quality of similarity classes and could give better clustering results.

Table 1: Clustering results of the first data set from CLUTO toolkit [7]

CLUTO			
Method	Entropy	Mutual information	*F* measure
Rb	0.562	0.261	0.641
Rbr	0.561	0.261	0.651
Direct	0.552	0.264	0.672
Agglo	1.283	0.001	0.452
Bagglo	0.455	0.299	0.712

Table 2: Evaluation of clustering results with SRSM and the new method for the first data set

α	SRSM					New method				
	Entropy	Mutual information	*F* measure	Size of similarity classes		Entropy	Mutual Information	*F* measure	Size of similarity classes	
				Max	Avg				Max	Avg
0.40	0.375	0.328	0.859	83	4.80	0.331	0.344	0.896	75	3.69
0.45	0.348	0.337	0.877	69	3.61	0.319	0.348	0.902	67	2.88
0.50	0.327	0.345	0.892	69	3.52	0.291	0.358	0.915	67	2.81
0.55	0.309	0.352	0.900	60	2.45	0.286	0.360	0.916	60	2.07
0.60	0.309	0.352	0.905	60	2.19	0.288	0.359	0.915	60	1.89
0.65	0.306	0.353	0.907	60	2.15	0.297	0.356	0.913	60	1.86
0.70	0.308	0.353	0.908	28	1.37	0.294	0.358	0.914	20	1.29
0.75	0.311	0.351	0.907	28	1.34	0.299	0.356	0.913	20	1.27
0.80	0.310	0.352	0.908	17	1.21	0.300	0.355	0.913	17	1.17

The maximum, the minimum and the average sizes of similarity classes of SRSM and the proposed method are shown in Table 2. Number of terms that are not in WordNet is 4753 among 13177 terms. It is around 36%.

We can see that sizes of similarity classes of the proposed method are smaller than those of SRSM. The difference is the result of removing terms with low WordNet based semantic relatedness from similarity classes in the proposed method. As defined by eq. (9), a similarity class of a term t_i consists of terms that satisfy both the condition of co-occurrence with t_i and one of the following conditions: 1) at least one of the two terms does not exist in WordNet database; 2) the WordNet based similarity measure between the two terms is greater than a threshold value. For example, when $\alpha = 0.55$, the average number of similarity classes defined only by the co-occurrence condition is 2.45 (SRSM), while the one defined by eq. (9) is 2.07 (the proposed method), which means that 0.38 terms in average are removed from similarity classes of SRSM because they do not satisfy the above condition 1) nor 2). Then, among the remaining 2.07 terms of similarity classes of the proposed method, 0.88 terms satisfy the condition 1) and 1.19 satisfy condition 2), in average.

The contingency table of the best case of the proposed method is shown in Table 3. Precision and recall of CACM, CISI, CRANFIELD, and MEDLINE are 0.964, 0.797, 0.930, 0.962 and 0.851, 0.952, 0.976, 0.983, respectively.

The computation time of the new method is almost same as the one of the SRSM method which has the time complexity of $O(M\log M)$ [7], where M is the number of terms in the text collection. The difference between the new and SRSM methods is the computation of term semantic relationship based on WordNet. The computation of semantic relationship is fast because we use a path based method and the maximum depth of the word hierarchy in WordNet is sixteen [9], a very small number in comparison with number of terms in a text collection.

Table 3: Contingency table of the best case of the proposed method

	CACM	CISI	CRANFIELD	MEDLINE
Cluster 1	2726	68	29	5
Cluster 2	347	1390	4	4
Cluster 3	94	0	1366	9
Cluster 4	37	2	1	1015

The second test collection used in our experiment is abstracts of papers from several IEEE journals of several fields. We formed a collection of 1010 documents from IEEE Transactions on Knowledge and Data Engineering (378 abstracts), IEEE Transactions on Biomedical Engineering (311 abstracts) and IEEE Transactions on Nanotechnology (321 abstracts). These categories of documents are denoted as KDE, BIO and NANO. We use the clustering methods to cluster the data set into three clusters.

After removing stopwords and stemming words, we have 5690 terms in the document collection. With 5690 terms, the algorithm created 1010 document vector using tf×idf weighting scheme, each document vector has 5690 dimensions.

Table 4: Clustering results of the second data set from CLUTO toolkit [7]

CLUTO			
Method	Entropy	Mutual information	F measure
rb	0.290	0.366	0.898
rbr	0.198	0.408	0.954
direct	0.198	0.408	0.954
agglo	0.684	0.187	0.723
graph	0.254	0.383	0.936
bagglo	0.234	0.392	0.939

Table 5: Evaluation of clustering results with SRSM and the new method for the second data set

α	SRSM			New method		
	Entropy	Mutual information	F measure	Entropy	Mutual information	F measure
0.30	0.205	0.405	0.953	0.133	0.438	0.971
0.35	0.141	0.434	0.970	0.125	0.442	0.974
0.40	0.155	0.428	0.965	0.122	0.443	0.975
0.45	0.175	0.418	0.960	0.137	0.436	0.971
0.50	0.179	0.417	0.959	0.150	0.430	0.967
0.55	0.172	0.420	0.962	0.186	0.414	0.956
0.60	0.182	0.416	0.956	0.161	0.425	0.963
0.65	0.196	0.408	0.952	0.174	0.419	0.959
0.70	0.202	0.406	0.951	0.188	0.413	0.955

Table 4 shows the evaluation of clustering results from CLUTO toolkit's algorithms. Table 5 shows the evaluation of clustering results of SRSM and the newly proposed method with different values of parameter α. The best evaluation in each quality measure values is shaded in Table 5.

For the WordNet semantic similarity based method, the evaluation of the clustering result was 0.363, 0.332, 0.894 for entropy, mutual information and *F* measure respectively.

The results show that clustering results of the newly proposed method are better than those of the other methods in all three evaluation measures.

6. Conclusions

The vector space model is widely used in the field of document clustering. It represents a document as a vector of terms. However, the simple VSM treats terms independent to each other and the semantic relationships between terms are not considered. Therefore, it reduces the effectiveness of document clustering methods. SRSM method and WordNet semantic similarity based method use the semantic relation between terms to improve the performance of document clustering. However, these methods have their own issues as we discussed in the previous sections. We proposed a new method that is a combination of SRSM and WordNet semantic similarity based method to solve these issues.

Our experiment results show that the quality of the clustering with the proposed method is better than the ones with SRSM and WordNet semantic similarity based method. Its clustering results are also better than results of other methods in the CLUTO toolkit.

In addition to WordNet, Wikipedia and Wiktionary are also promising tools for semantic relatedness measurement and analysis [22]. In our future work, we will exploit these tools to further improve document clustering methods.

References

[1] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets", Data Mining and Knowledge Discovery , 10 (2), pp. 141 - 168, 2005.

[2] I.S. Dhillon and D.S. Modha, "Concept decompositions for large sparse text data using clustering", Machine Learning, 42 (1-2), pp. 143-175, 2001.

[3] M. Steinbach, G. Karypis and V. Kumar, "A comparison of document clustering techniques", Proceedings of the KDD Workshop on Text Mining, 2000.

[4] Y. Li, S.M. Chung and J.D. Holt, "Text document clustering based on frequent word meaning sequences", Data and Knowledge Engineering, 64 (1), pp. 381-404, 2008.

[5] M. Mahdavi and H. Abolhassani, "Harmony K-means algorithm for document clustering", Data Mining and Knowledge Discovery, pp. 1-22, 2008.

[6] T.B. Ho and K. Funakoshi, "Information retrieval using rough sets", Journal of Japanese Society for Aritificial Intelligence, 13 (3), pp. 424-433, 1997.

[7] N.C. Thanh, K. Yamada and M. Unehara, "A Similarity Rough Set Model for document representation and document clustering", Journal of Advanced Computational Intelligence and Intelligent Informatics, 15 (2), pp. 125-133, 2011.

[8] W.K. Gad and M.S. Kamel, "Enhancing text clustering performance using semantic similarity", Lecture Notes in Business Information Processing, 24 LNBIP, pp. 325-335, 2009.

[9] L. Jing, M.K. Ng and J.Z. Huang, "Knowledge-based vector space model for text clustering", Knowledge and Information Systems, 25 (1), pp. 35-55, 2010.

[10] Z. Pawlak, "Rough sets", Int. J. of Information and Computer Sciences, 11 (5), pp. 341-356, 1982.

[11] R. Slowinski and D. Vanderpooten, "Similarity Relation as a basis for rough approximation", Advances in Machine Intelligence and Soft Computing, Vol.4, pp. 17-33, 1997.

[12] R. Slowinski and D. Vanderpooten, "A generalized definition of rough approximations based on similarity", IEEE Trans. on Knowledge and Data Engineering, 12 (2), pp. 331-336, 2000.

[13] T.B. Ho and N.B. Nguyen, "Nonhierarchical document clustering based on a tolerance rough set model", International Journal of Intelligent Systems, 17 (2), pp. 199-212, 2002.

[14] X.-J. Meng, Q.-C. Chen, and X.-L. Wang, "A tolerance rough set based semantic clustering method for web search results", Information Technology Journal, 8 (4), pp. 453-464, 2009.

[15] Princeton University, "About WordNet", WordNet, Princeton University. 2010, http://wordnet.princeton.edu.

[16] R. Rada, H. Mili, E. Bicknell and M. Blettner, "Development and application of a metric on semantic nets", IEEE Transactions on Systems, Man and Cybernetics, v (n), pp. 17-30, 1989.

[17] Z. Wu and M. Palmer, "Verbs semantics and lexical selection", Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL '94), pp. 133-138, 1994.

[18] P. Resnik, "Using information content to evaluate semantic similarity", Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 448-453, 1995.

[19] J. J. Jiang and D. W. Conrath. “Semantic similarity based on corpus statistics and lexical taxonomy”, Proceedings of the 10th International Conference on Research in Computational Linguistics, Taipei, Taiwan, 1997.
[20] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone”, Proceedings of the 5th Annual International Conference on Systems Documentation, pp. 24-26, 1986.
[21] S. Banerjee and T. Pedersen, “An adapted lesk algorithm for word sense disambiguation using WordNet”, CICLing ' 02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, pp. 136-145, 2002.
[22] T. Zesch and I. Gurevych, “Wisdom of crowds versus wisdom of linguists - Measuring the semantic relatedness of words”, Natural Language Engineering, 16 (1), pp. 25-59, 2010.
[23] S. Patwardhan and T. Pedersen, “Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts”, Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together, pp. 1-8, Trento, Italy, 2006.
[24] G. Karypis, “CLUTO - A Clustering Toolkit”, 2003, http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download.
[25] ftp://ftp.cs.cornell.edu/pub/smart
[26] A. Strehl, J. Ghosh and R. Mooney, “Impact of similarity measures on web-page clustering”, Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web search (AAAI 2000), Austin, TX, pp. 58–64, July 2000.