Textual Knowledge Representation through the Semantic-based Graph Structure in Clustering Applications

Jiangning WU, Yanzhong DANG, Donghua PAN, Zhaoguo XUAN, Qiaofeng LIU
Institute of Systems Engineering, Dalian University of Technology, Dalian, China
{jnwu, yzhdang, gyise}@dlut.edu.cn

Abstract
To represent the textual knowledge more expressively, a kind of semantic-based graph structure is proposed for this issue and thereafter applied to clustering problems. Such graph structure for textual representation consists of nodes and directed edges, which stand for the feature terms derived from the texts and the semantic relationships between them, respectively. Moreover, the weight is assigned to each edge so that the strength of relationship between two terms can be measured. For this weighted directed graph structure, a novel graph similarity algorithm is developed by extracting the maximum common subgraph between two concerned graphs, which can therefore be used to measure the distance between two graph structures, i.e. two texts, and finally be used to sort the texts into different clusters. Some experiments have been done through the proposed semantic graph structure in clustering applications and the results have proved the high performance of our textual knowledge representation model.

1. Introduction
Knowledge representation is the initial step in the knowledge management process on an individual as well as on an organizational level. With respect to textual knowledge discovery and representation, they nowadays have become the popular research topics since text is the most common form of knowledge storage. The underlying problem with the textual representation is the expressivity of semantic information of the text. Typical model like the vector space model (VSM) is simple and allows the application of traditional methods that deal with numerical feature vectors in a Euclidean feature space. However, the traditional paradigm in these kinds of models has discarded the important semantic and structural information when the original text is converted to a vector of numerical values. Considering that graphs are the strong mathematical constructs and can model relationships and structural information effectively, they are accordingly adopted in our study.

There are various forms of graphs, such as tree, network, and so forth, where the network-like structure can better reflect both context and content information of the text, ad hoc syntactic information (e.g. phrase structure, word order, proximity information) [1][2]. From this point of view, a novel, network-like, semantic-based graph structure model for textual knowledge representation is proposed, whose contributions lie in the following aspects.

First, the directed edges of the graph are defined based on the position information of terms that occur in the same sentence together, aimed at to keep more word order information in the graph.

Second, the weight is assigned to each edge of the graph to measure the strength of relationship between a pair of terms. Finally, a directed weighted semantic-based graph can be obtained.

Third, a new graph similarity measure is designed by introducing a maximum common subgraph of any two graphs which contains more structural information than the numerical feature vector does during the similarity calculation.

In this way, more semantic and order information among terms as well as the structural information of sentences can be maintained in the proposed semantic-based graph structure.

Some experiments have been done through our weighted directed graph structure and the corresponding similarity algorithm in clustering applications, and the results have proved the high performance of our textual knowledge representation model.

2. A semantic-based directed weighted graph structure model

In this section, several definitions about the graph are given at first. Then the detailed procedure for constructing a graph structure model from the text is described. At last, a formula for calculating the
distance between a pair of directed weighted graphs is put forward.

2.1. Definitions

First of all, we give some definitions about graphs for further understanding.

- **Graph**: Let a triplet \( G=(N, E, W) \) be the weighted graph, where \( N \) is a set of nodes \( N=\{n_1, n_2, ... , n_k\} \), \( E \) is a set of directed edges connecting the nodes, \( E=\{e(n_i, n_j)| n_i \subseteq N, n_j \subseteq N, i,j \subseteq [1,k], i \neq j\} \), and \( W \) is a set of weights assigned to the current edge. For the whole document set, a graph set can be obtained denoted as \( G=\{g_1, g_2, ... , g_i, ... \} \). The pseudocode of the directed weighted graph generation algorithm is given below.

| Step 1: | Initialize node set \( N=\{\} \), edge set \( E=\{\} \) and weight set \( W=\{\} \). And map the elements \( t_i(i=1, ..., m) \) of term set \( T_i=\{t_1, freq_1\}, \{t_2, freq_2\}, ... , \{t_m, freq_m\} \) as the nodes of the graph, \( e \) indicates the directed edges between a pair of terms and \( w \) implies the weight assigned to the current edge. For the whole document set, a graph set can be obtained denoted as \( G=\{g_1, g_2, ... , g_i, ... \} \). The pseudocode of the directed weighted graph generation algorithm is given below.

| Step 2: | Define a unit for a given sentence.

| Step 3: | Randomly choose a pair of nodes \( n_i \) and \( n_j \) from the initial node set \( n_0 \). If the corresponding terms \( t_i \) and \( t_j \) appear together in the current unit, then construct a directed edge from \( n_i \) to \( n_j \) where \( t_i \) is ahead of \( t_j \). And add these two nodes to the node set \( N=\{n_i, n_j\} \).

| Step 4: | Count the times for \( t_i \) and \( t_j \) appearing in the unit respectively as well as the times \( freq(t_i, t_j) \) for \( t_i \) and \( t_j \) appearing together in the unit.

| Step 5: | Compute the weight \( w_{ij} \) on the directed edge \( e(n_i, n_j) \).

| Step 6: | Repeat steps 2 to 5 until the directed weighted graph completes.

Each step of the algorithm is explained in more detail as follows.

In step 1, all the feature terms are mapped to the graph as the initial nodes. After checking the relationships between feature terms, some of them with zero or lower co-occurrence frequencies are removed from the future graph.

In step 2, a minimum length of a sentence as a unit is selected to measure the co-occurrence information of feature terms instead of a whole paragraph or only two adjacent terms. The reason why we don’t use the paragraph as the unit lies that more term pairs appearing together can be found there, which will result in a bigger graph and meanwhile weaken the effect of mutual information of feature terms. With respect to the adjacent terms, if only adjacent terms are selected as a unit like Schenker’s model [1], there

2.2. The directed weighted graph generation

Representing texts with graph structures can retain additional information such as the inner structures, some semantic relationships and the orders of terms in the texts all of which can not be captured by using the traditional vector representations. In our model, we utilize the directed weighted graph structure to represent the text.

Suppose a document set \( D=\{d_1, d_2, ... , d_n\} \) is given. After preprocessing, a set of feature terms \( T=\{t_{d_1}, t_{d_2}, ... , t_{d_n}\} \) can be arrived at, where \( T_{d_i} \) is a collection of feature terms in \( d_i \). In the study, we use a binary set \( T_{d_i}=\{[t_1, freq_1], [t_2, freq_2], ... , [t_m, freq_m]\} \) composed of feature terms and their frequencies to record the times of feature terms appearing in the texts, where \( freq \) is the frequency of term \( t_i \) appearing in \( d_n \), and \( m \) is the number of feature terms. Meanwhile, the position of each feature term is recorded either for the further use. By utilizing the selected feature terms together with their directed relationships, a directed graph \( g=\{n, e, w\} \) related to the given text can be built, in which \( n \) represents the node of the graph, \( e \) indicates the directed edges between a pair of terms and \( w \) implies the weight assigned to the current edge.
would not be enough mutual information of terms to be used due to the strict limitation.

In step 3, only the terms appearing together in a given unit can be selected as the nodes of the graph. Then a directed edge can be added to link these two nodes based on their order information.

In steps 4 and 5, if terms appear together in the units with a higher frequency, it means there is a compact relation between them. The link should be stronger and a larger weight should be assigned to it.

The formula for evaluating the strength of the relation between terms \( t_i \) and \( t_j \) is given below.

\[
w_{ij} = \frac{freq(t_i, t_j)}{freq(t_i) + freq(t_j) - freq(t_i, t_j)}
\]

where \( w_{ij} (i, j = 1, 2, \ldots, m) \) denotes the weight on the directed edge connecting \( n_i \) and \( n_j \), \( freq(t_i, t_j) \) is the times for \( t_i \) and \( t_j \) appearing together in the unit, \( freq(t_i) \) and \( freq(t_j) \) denote the frequencies of terms \( t_i \) and \( t_j \) appearing in \( d_i \) respectively. The high \( w_{ij} \) corresponds to a strong link between terms and the low \( w_{ij} \) to a weak link.

In step 6, repeatedly perform the above steps, and finally the whole graph structure model can be built for representing the text.

Figure 1 shows an automatically generated semantic-based graph structure from a text [3] by the Netdraw software [4], in which the selected feature terms and the corresponding directed edges with different weights are demonstrated. The nodes of the graph are the feature terms in the given text after preprocessing. If two feature terms occur together in the units, there will be the directed edge connected them. The weight allocated to the current edge implies the link stronger or not.

For the sake of clearness, a weight threshold, say 0.3, is performed to the original graph and a simplified graph structure is obtained accordingly, as shown in Figure 2.

**Figure 2.** A simplified graph structure by using a weight threshold (the edge weight = 0.3)

### 2.3. Graph distance measuring

Graph distance is a numeric measure of dissimilarity between graphs, with larger distances implying more dissimilarity which is usually used in fields of image processing, chemistry and pattern recognition [5][6][7]. In our research, we utilize the distance between maximum common subgraphs to measure the similarity of texts represented by the directed weighted graphs.

#### 2.3.1. Mcs-based similarity calculation algorithm.

The process for extracting the maximum common subgraph (mcs) from a pair of graphs is very complicated. Some approaches can be found in references [8][9][15][16]. The general way is to create a compatibility graph for the two given graphs, and then to find the largest graph within them. Unfortunately the computation involved is a NP-complete problem. In our approach, we don’t follow the definition of mcs in a mathematical way, and simply employ the information on nodes, directed edges and weights of the graph to extract the mcs. Concretely, the maximum common subgraph, \( G_{mcs} \), of a pair of graphs \( G_1 \) and \( G_2 \) can be created by the following procedure.

1) Find the common nodes \( n_{mcs} \) by determining the subsets of nodes that are contained both in the original graphs \( G_1 \) and \( G_2 \).

2) Find the directed edges by examining all pairs of nodes extracted from the previous step and keep the common directed edges \( e_{mcs} \) that are contained both in the original graphs \( G_1 \) and \( G_2 \).
3) Compare the weights $w_{mcs}$ on the directed edges between two original graphs and hold the smaller ones.

Unlike the model in [10] in which only distance measure was introduced to evaluate the size of mcs, i.e. the similarity of two graphs, instead a novel algorithm is proposed in our approach to compute the similarity of two graphs by considering the contributions both from the common nodes and from the common edges as well as their weights. The distance of graphs $G_1$ and $G_2$ can be calculated by the following formulas:

$$\text{Distance}(G_1, G_2) = 1 - \text{sim}(G_1, G_2)$$ (2)

$$\text{sim}(G_1, G_2) = \beta \frac{|N(g)|}{\max(|N(G_1)|, |N(G_2)|)} + (1 - \beta) \sum \forall E(g)(\min(w_{ij}, w_{ij}')) / \max(w_{ij}, w_{ij}')$$ (3)

where $g = \text{mcs}(G_1, G_2)$ denotes the mcs of $G_1$ and $G_2$; $|N(g)|$ is the number of nodes in $g$ and $E(g)$ is the number of edges in $g$; $w_{ij}$ and $w_{ij}'$ denote the weights of $e_{ij}$ in $G_1$ and of $e_{ij}'$ in $G_2$ respectively; $\max(|N(G_1)|, |N(G_2)|)$ is the larger number of nodes in $G_1$ or $G_2$; $\beta$ is an artificial coefficient determined by the user, and $\beta \in (0,1)$.

### 2.3.2. Time complexity analysis.

Assume that there are two graphs $G_1$ and $G_2$ with $|n_1|$ and $|n_2|$ nodes respectively, and their mcs contains $|n_{mcs}|$ nodes. In the process of finding common nodes from two graphs, the time complexity is $O(|n_1|\times|n_2|)$.

In the process of finding common directed edges, the time complexity is $O(|n_{mcs}|^2)$ obtained from Formula (4).

$$C_{mcs}^2 = \frac{|n_{mcs}| - \left(\frac{|n_{mcs}|}{2}\right) - 1}{2}$$ (4)

$$= \frac{|n_{mcs}|^2 - |n_{mcs}| - 1}{2}$$

Among $|n_{mcs}|$ nodes, the time complexity for finding the directed edges is $O(|n_1|^2 \times |n_2|^2 + |n_{mcs}|^2) < O(|n_1|^2 + |n_{mcs}|^2) = O(|n|^2)$ by substituting $|n| = \max(|n_1|, |n_2|)$.

### 3. Clustering applications based on the proposed model

Traditional clustering with graphs treats nodes as the items to be clustered and weights on edges as the distance between objects the nodes represent. The usual procedure is to create a minimum spanning tree of the graph and then remove the remaining edges with the largest weight in the minimum spanning tree until the number of desired clusters (connected components) is achieved [11][12], where the connected components indicate which objects belong to which clusters and objects whose nodes are connected by edges are in the same cluster. This kind of method often employs the graph edit distance to measure the similarity of objects. One drawback of this approach is that the edit cost functions must be specified for each application. Therefore, the simple and effective graph-based clustering algorithm is called for.

In our study, we extend the classical and simple $k$-means clustering algorithm to make it suitable for the developed directed weighted graph structure. The extension contains two aspects: 1) Convert the traditional vector-based distance computation into the graph structure based distance measure; 2) Transform the calculation of distance between feature terms and the cluster center into the graph structure to the cluster center. The extended $k$-means clustering algorithm is given below.

#### Extended $k$-means clustering algorithm

**Input:** a set of graph $G = \{G_1, G_2, \ldots, G_m\}$, and the parameter $k$ defining the number of clusters to be created.

**Output:** clusters to which the directed weighted graphs $G(i \in [1,n])$ belong, and the clustering centers $M(j \in [1,k])$.

**Procedure:**

**Step1.** Randomly assign $m$ graphs to $k$ clusters $C(j \in [1,k])$;

**Step2.** Compute the cluster center $M(j \in [1,k])$ of each cluster $C(j \in [1,k])$;

**Step3.** Compute the distance between each graph $G(i \in [1,n])$ in graph set $G = \{G_1, G_2, \ldots, G_m\}$ and the cluster center $M(j \in [1,k])$. Put $G(i \in [1,n])$ into the nearest cluster, and then get a new cluster $C'(j \in [1,k])$;

**Step4.** Re-compute the cluster center $M'(j \in [1,k])$ of each new cluster $C'(j \in [1,k])$;

**Step5.** Repeat steps 2, 3 and 4 until the cluster center $M'(j \in [1,k])$ has no change.

The clustering center involved in the above expended $k$-means clustering algorithm can be calculated by the following formula:

$$M = \arg \max_{g \in G} \left( \frac{1}{m} \sum_{i=1}^{m} \text{sim}(g, g_i) \right)$$ (5)

where $G$ is a set of graphs, $m$ is the number of graphs in $G$, and $g_i$ is a graph in $G$. 

4. Experiments

By using the proposed semantic-based graph structure model described in Section 2, we can generate, for each text, a directed weighted graph. Thereafter an extended $k$-means clustering algorithm is performed for clustering tasks.

4.1. Dataset

Chinese corpus from the natural language processing group of Fudan University in China is selected as the data analyzed. It is open and available at http://www.nlp.org.cn/docs/download.php?doc_id=295, including nearly twenty thousand pieces of texts with 20 categories. The reason why we choose Chinese texts is the speciality of the language itself. Apart from the counterpart problem in English, Chinese has its own characteristics, such that there is no space between words in sentences and there are 20,000 to 50,000 words frequently used in Chinese, which are much more than the number of words used in English [13].

Owing that Chinese words do not have a remarkable boundary which is greatly different from most of the western languages, the word segmentation is necessary before any other preprocessing. Here we perform a segmenting method without the dictionary to extract the candidate feature terms from the original texts [14]. The other tasks including stop-word removing and synonym combination in preprocessing are also fulfilled by using the predefined stop-word list and an open electronic dictionary (http://www.ir-lab.org/) to deal with the synonyms.

For reducing the computing time, we choose 5 categories from the corpora on different topics (environment, computer, agriculture, politics and sports) which are randomly divided into the training text set and the testing text set. The average length of the selected documents in the data set contains about 3,600 words.

4.2. Experiment setup

In order to evaluate the performance of the proposed graph structure and the improved clustering algorithm, three measures are introduced, i.e. precision, recall and $F$-Score. The $F$-Score measurement is

$$F(r, p) = \frac{2rp}{r + p}$$

where $p$ and $r$ respectively are the precision and recall of the clustering algorithm.

The experiment we have conducted consists of the following four steps: firstly, randomly choose 20 pieces of texts from each category of the corpus, and get total 100 pieces of texts from 5 categories; secondly, select the feature terms from the texts and construct the directed weighted graph structures; thirdly, utilize the mcs-based distance measure to gain the similarities between graph structures; at last, all similarities of the chosen 100 pieces of texts can be obtained, which can then be used for further clustering with the downward hierarchical clustering analysis (DHCA) algorithm and the extended $K$-means algorithm respectively.

4.3. Experimental results

In the experiment, we set $k=5$ and $\beta=0.6$, the similarity threshold is 0.02. For the graphs whose similarities below 0.02, they will be ignored in order to reduce the running time. After 20-time running, the steady clustering results are gained, as shown in Table 1 and Figure 3, in which $T$ denotes the number of classified texts, $F_i$ denotes the number of correctly classified texts, and Orig. is original category for short.

After clustering by means of the extended $k$-means clustering algorithm, 5 clusters $N-C19$, $N-C31$, $N-C32$, $N-C38$ and $N-C39$ with five cluster labels Computer, Environment, Agriculture, Politics and Sports are obtained. The isolated graphs $C19$-Computer0776 and $C38$-Politics0026 can’t be grouped into any categories because they have great distance to the other graphs.

We also compare the clustering results obtained from two different clustering algorithms, i.e. the extended $k$-manes algorithm and the DHCA algorithm, as shown in Table 2 and Figure 4. Figure 4 indicates that clusters Computer and Environment have a high precision, and the cluster Politics has a low precision this is because the texts in this cluster have mixed with several texts in Sports. As a result, the recall of the cluster Sports is not very high.
Table 1. Statistics of clustering results

<table>
<thead>
<tr>
<th>Orig. Category</th>
<th>Orig-C19</th>
<th>Orig-C31</th>
<th>Orig-C32</th>
<th>Orig-C38</th>
<th>Orig-C39</th>
<th>T</th>
<th>F</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-C19</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>100.00</td>
<td>88.89</td>
<td>0.9412</td>
</tr>
<tr>
<td>N-C31</td>
<td>0</td>
<td>16</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>1</td>
<td>94.12</td>
<td>88.89</td>
<td>0.9143</td>
</tr>
<tr>
<td>N-C32</td>
<td>0</td>
<td>1</td>
<td>17</td>
<td>1</td>
<td>2</td>
<td>17</td>
<td>4</td>
<td>80.95</td>
<td>89.47</td>
<td>0.8500</td>
</tr>
<tr>
<td>N-C38</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>3</td>
<td>16</td>
<td>5</td>
<td>76.19</td>
<td>94.12</td>
<td>0.8421</td>
</tr>
<tr>
<td>N-C39</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>2</td>
<td>86.67</td>
<td>72.22</td>
<td>0.7879</td>
</tr>
</tbody>
</table>

Figure 3. Statistical analysis of the clustering results

Table 2. Measuring values with two different clustering algorithms

<table>
<thead>
<tr>
<th>Evaluation Measures</th>
<th>New Category</th>
<th>Extended k-means clustering algorithm</th>
<th>DHCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision (%)</td>
<td>Recall (%)</td>
</tr>
<tr>
<td>N-C19</td>
<td></td>
<td>100.00</td>
<td>88.89</td>
</tr>
<tr>
<td>N-C31</td>
<td></td>
<td>94.12</td>
<td>88.89</td>
</tr>
<tr>
<td>N-C32</td>
<td></td>
<td>80.95</td>
<td>89.47</td>
</tr>
<tr>
<td>N-C38</td>
<td></td>
<td>76.19</td>
<td>94.12</td>
</tr>
<tr>
<td>N-C39</td>
<td></td>
<td>86.67</td>
<td>72.22</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td>87.59</td>
<td>86.72</td>
</tr>
</tbody>
</table>
Moreover, we can see from Figure 4 that the clustering results obtained from the extended $k$-means algorithm are better in terms of three categories than from the DHCA algorithm. The maximum increment of precision of the extended $k$-means clustering algorithm is up to 10% (see C31-Environment).

5. Conclusion

In this paper we present a novel textual knowledge representation model by using a semantic-based directed weighted graph structure, in which more semantic and order information among terms as well as the structural information of the text are stored. Such model can be constructed by extracting representative terms from texts and their mutually semantic relationships. By introducing the proposed model into the extended $k$-means clustering algorithm, the good clustering results have been obtained in terms of precision, recall and F-score. It turns out that the developed graph structure model has a satisfied representation capability and the extended $k$-means clustering algorithm has a higher performance than DHCA.

Acknowledgements

This work is sponsored by the National Natural Science Foundation of China (NSFC) under grant No. 70771019 and supported by the National High Technology Research and Development Program of China (No.2008AA04Z107).

References

[14] S. H. Jiang, Y. Z. Dang, and Z. G. Xuan, Comparative
