ENHANCING TEXT REPRESENTATION FOR CLASSIFICATION TASKS WITH SEMANTIC GRAPH STRUCTURES

JIANGNING WU, ZHAOGUO XUAN AND DONGHUA PAN

Institute of Systems Engineering
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116023, P. R. China
{jinwu; gyise}@dlut.edu.cn; xzg@dl.cn

Received January 2010; revised June 2010

ABSTRACT. To represent the textual knowledge more expressively, a kind of semantic-based graph structure is proposed, in which more semantic and ordering information among terms as well as the structural information of the text are incorporated. Such model can be constructed by extracting representative terms from texts and their mutually semantic relationships. Afterward, it is represented as a graph, whose nodes are the selected terms and whose edges are the corresponding relationships respectively. Moreover, the weight is assigned to each edge so that the strength of relationship between two terms can be measured. Furthermore, 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 further be applied to classification tasks. Finally, some experiments have been conducted with the Chinese benchmark corpus for validation. The experimental results have proved the better performance of the proposed textual knowledge representation model in terms of its precision and recall.

Keywords: Text representation, Graph structure, Maximum common subgraph, Classification

1. Introduction. Text representation is the essential step for the tasks of text mining, such as text clustering, text classification and so on. It nowadays has become one of the popular research topics in text mining since text is the most common form of information storage. One of the underlying problems with the textual representation is the expressivity of semantic information in the texts. Typical model like the vector space model (VSM) [1] is simple and only 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 trees, networks and so forth, where the network-like structure can better reflect both contextual and semantic information of the text, and ad hoc syntactic information (e.g., phrase structure, word order, proximity information). The idea of graph representation for web content was originally presented in [2] and has been applied to such fields as symbolic images [13], document retrieval [14], etc. In [2], they viewed terms of HTML documents as nodes of the graph and relationships between terms as edges, thus a graph structure representation model for web content can be built. Motivated by the previous work [2-4, 13-16], the paper presents a novel weighted graph-structure model, in which more semantic and ordering information
among terms as well as the structural information of the text can be incorporated. The primary contribution of this study is to provide a more expressive way to represent the texts, especially for Chinese texts, towards such applications as text classification, text clustering, etc. More specially, the following work has been undertaken in this study:

- Represent the text as a graph structure by viewing the selected terms from the text as nodes and the co-occurrence relationships of terms as edges. Thereafter the directed edges of the graph are defined based on the position information of terms that occur in the same unit together. As a result, more word order information in the graph can be retained;
- Allocate each edge of the graph a weight coming from the related term frequency information. Thus an edge-weighted graph structure is built to measure the strength of relationship between a pair of terms;
- Introduce a maximum common subgraph (mcs) of two graphs for graph similarity measure, which contains more structural information than the numerical feature vector does during the similarity calculation.

Several experiments on the proposed network-like, semantic-based graph-structure model for textual representation have been undertaken with the Chinese benchmark corpus for validation. The experimental results indicate that the proposed text classification algorithm, based on the weighted semantic graph structure, performs well in terms of its precision and recall in Chinese context.

The remaining sessions of this paper are organized as follows. In Section 2, a semantic graph-structure model for textual representation have been undertaken with the Chinese benchmark corpus for validation. The experimental results indicate that the proposed text classification algorithm, based on the weighted semantic graph structure, performs well in terms of its precision and recall in Chinese context.

The remaining sessions of this paper are organized as follows. In Section 2, a semantic graph-structure model for textual representation have been undertaken with the Chinese benchmark corpus for validation. The experimental results indicate that the proposed text classification algorithm, based on the weighted semantic graph structure, performs well in terms of its precision and recall in Chinese context.

The remaining sessions of this paper are organized as follows. In Section 2, a semantic graph-structure model for textual representation have been undertaken with the Chinese benchmark corpus for validation. The experimental results indicate that the proposed text classification algorithm, based on the weighted semantic graph structure, performs well in terms of its precision and recall in Chinese context.

2. Semantic Graph Structure of the Text. 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, which cannot be captured by using the traditional vector representations. Especially in our model, we define a weighted semantic graph to represent the text. The nodes in the graph represent the terms involved in the text, and the directed edges represent the co-occurrence relationship of two terms. The direction of the edges records the ordering information of co-occurrence terms and the weight on each edge implies the frequent status of co-occurrence terms.

2.1. Two definitions. Before constructing the graph-structure model, two preliminary definitions are presented below in advance.

**Definition 2.1.** Graph: Let a triplet \( G = (N, E, W) \) be the weighted graph, where \( N \) is a set of nodes \( N = \{n_1, n_2, \ldots, n_j, \ldots, n_k\} \), \( E \) 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 related edges. The weight \( w_{ij} \) reflects the mutual information between nodes.

**Definition 2.2.** Unit: Let the minimum length of one sentence be a unit, where minimum length indicates different parts of one sentence separated by different punctuations such as comma, colon, semi-colon and stop. Meanwhile, it should cover at least two words.

Suppose a document set \( D = \{d_1, d_2, \ldots, d_i, \ldots\} \) is given. After preprocessing, a set of feature terms \( T = \{T_{d_1}, T_{d_2}, \ldots, T_{d_i}, \ldots\} \) can be extracted, where \( T_{d_i} \) is a collection of feature terms in document \( d_i \). In the study, we adopt a binary set \( T_{d_i} = \{[t_1, freq(t_1)], \ldots\} \),
Step 1: Initialize the node set \( N = \{ \} \), edge set \( E = \{ \} \) and weight set \( W = \{ \} \). And map the elements \( t_i \) (\( i = 1, \ldots, m \)) of term set \( T_d = \{ [t_1, \text{freq}(t_1)], [t_2, \text{freq}(t_2)], \ldots, [t_i, \text{freq}(t_i)], [t_j, \text{freq}(t_j)], \ldots, [t_m, \text{freq}(t_m)] \} \) as the nodes of the graph and generate an initial node set \( n_0 = \{ n_1, n_2, \ldots, n_i, n_j, \ldots, n_m \} \);  

Step 2: Define a unit (see Definition 2.2) 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 \( t_j \). Meanwhile 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 \( \text{freq}(t_i, t_j) \) 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 in the algorithm is addressed and explained in detail as follows.

In Step 1, all the feature terms are mapped into 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 do not use the paragraph as the unit lies in that more term pairs appearing together can be found there, which will result in a larger 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 [2], there would not be sufficient 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 close 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.

\[
  w_{ij} = \frac{\text{freq}(t_i, t_j)}{\text{freq}(t_i) + \text{freq}(t_j) - \text{freq}(t_i, t_j)} \tag{1}
\]

where \( w_{ij} \) (\( i, j = 1, 2, \ldots, m \)) denotes the weight on the directed edge connecting \( n_i \) and \( n_j \), \( \text{freq}(t_i, t_j) \) is the times for \( t_i \) and \( t_j \) appearing together in the unit, and \( \text{freq}(t_i), \text{freq}(t_j) \) denote the frequencies of terms \( t_i \) and \( t_j \) appearing in \( d_i \) respectively. Moreover, the high \( w_{ij} \) corresponds to a strong link between terms and the low \( w_{ij} \) to a weak link.
In Step 6, after repeatedly perform the aforementioned steps, finally, the whole graph structure for representing the text can be built, as shown in Figure 1.

Figure 1 shows an automatically generated semantic-based graph structure for a text [5] by the Netdraw software [6], 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 a directed edge connecting them. The weight allocated to the current edge implies whether the link is stronger or not. For the sake of clearness, a weight threshold, say 0.3, is applied to the original graph. Then a simplified graph structure is obtained accordingly, as shown in Figure 2.
3. Mcs-based Similarity Calculation Algorithm. In terms of the differences between graph structure and conventional word vector, it is natural to content that the traditional similarity calculation algorithms, e.g., cosine similarity measures, inner product similarity measures, are no longer appropriate for calculating the similarities between two graphs. Therefore, in this study, a new maximum-common subgraph based algorithm is proposed for calculating the similarities between weighted graphs.

3.1. Basic definitions.

Definition 3.1. Subgraph: Let a graph \( G_1 = (N_1, E_1, W_1) \) be a subgraph of a graph \( G_2 = (N_2, E_2, W_2) \), denoted as \( G_1 \subseteq G_2 \), for \( N_1 \subseteq N_2, E_1 \subseteq E_2 \cap (N_1 \times N_1) \). Conversely, the graph \( G_2 \) is called a supergraph of \( G_1 \).

Definition 3.2. Maximum-common subgraph: A graph \( g \) is a maximum-common subgraph \( mcs \) of graphs \( G_1 \) and \( G_2 \), denoted as \( mcs(G_1, G_2) \), if \( g \subseteq G_1, g \subseteq G_2 \) and there is no other subgraphs \( g' \) (\( g' \subseteq G_1, g' \subseteq G_2 \)) such that \( |g'| > |g| \).

Definition 3.3. Similarity between two graphs: For two graphs \( G_1 \) and \( G_2 \), the similarity between them means the degree they look like with each other, denoted as \( sim(G_1, G_2) \). It is a function that has the following properties [2]:

1. \( 0 \leq \text{sim}(G_1, G_2) \leq 1 \).
2. \( \text{sim}(G_1, G_2) = 1 \rightarrow G_1 \cong G_2 \).
3. \( \text{sim}(G_1, G_2) = \text{sim}(G_2, G_1) \).
4. if \( G_1 \) is more similar to \( G_2 \) than to \( G_3 \), then \( \text{sim}(G_1, G_2) \geq \text{sim}(G_1, G_3) \).

3.2. Mcs-based similarity calculation algorithm. The process for extracting the maximum common subgraph from two graphs is very complicated. Some approaches can be found in [8-10]. The general way is to start with creating a compatibility graph for two given graphs, and then to get the largest graph within them. Unfortunately, the computation involved in this way is a NP-complete problem. Taking this into consideration, in our approach, we do not follow the definition of \( mcs \) in a mathematical way. Instead, we simply employ the information on nodes, directed edges and weights of the graph to extract the \( mcs \). More specially, the maximum common subgraph, \( g_{mcs} \), of a pair of graphs \( G_1 \) and \( G_2 \), can be created by the procedure below:

First, find the common nodes \( n_{mcs} \) by determining the subsets of nodes that are contained both in the original graphs \( G_1 \) and \( G_2 \);

Second, find the edges by examining all pairs of nodes extracted from the previous steps and keep the common edges \( e_{mcs} \) that are contained both in the original graphs \( G_1 \) and \( G_2 \);

Third, compare the weights \( w_{mcs} \) on the edges between two original graphs and hold the smaller ones.

Unlike the model in [3] in which only distance measure was introduced to evaluate the size of \( mcs \), i.e., the similarity of two graphs, the novel algorithm proposed in our approach is 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 similarity between graphs \( G_1 \) and \( G_2 \) can be calculated by the following formula:

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

(2)

where \( g = mcs(G_1, G_2) \) 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 weight of \( e_{ij} \) in \( G_1 \) and the
weight 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 $\beta \in (0, 1)$.

3.3. **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. It is not hard to get that the time complexity is $O(|n_1| \cdot |n_2|)$ in the process of finding common nodes from two graphs. Furthermore, in the process of finding common edges, the time complexity is $O(|n_{mcs}|^2)$, acquired from Formula (3) below.

$$C_{|n_{mcs}|}^2 = \frac{|n_{mcs}| \cdot (|n_{mcs}| - 1)}{2} = \frac{|n_{mcs}|^2 - |n_{mcs}|}{2} < |n_{mcs}|^2$$

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

4. **Experiments.** To testify the representation capabilities of the proposed model, we design and conduct several experiments for text classification tasks.

4.1. **Dataset.** The data source we employ comes from the natural language processing group of Fudan University in China, which is publicly released (available at http://www.nlp.org.cn/docs/download.php?doc_id=295). The corpora contains 20 categories and more than twenty thousand pieces of texts. The reason why we choose Chinese texts lies in the speciality of the language itself. Apart from the counterpart problem in English, Chinese has its own characteristics, e.g., 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 frequently used in English [11].

Since 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. Especially, in this study, we perform a segmenting method without the dictionary to extract the candidate feature terms from the original texts [12]. The other tasks, including stop-word removing and synonym combination in preprocessing, are also fulfilled using the predefined stop-word lists and an open electronic dictionary (http://www.ir-lab.org/).

4.2. **Experiment setup.** To classify the selected text set, the following formula is deployed:

$$S(d_i, c_j) = \frac{1}{q} \sum_{d_{jp} \in c_j} \text{sim}(d_i, d_{jp})$$

where $d_i$ denotes the text to be classified, $c_j$ represents the class label; $S(d_i, c_j)$ denotes the average similarity between the text $d_i$ and the texts in class $c_j$; $d_{jp}$ ($p = 1, \ldots, q$) denotes the text collections in class $c_j$; $q$ is the number of texts in class $c_j$; and $\text{sim}(d_i, d_{jp})$ denotes the similarity between the text $d_i$ and the other texts in class $c_j$. Formula (4) can calculate the similarities between the text $d_i$ and the other texts belonging to any class $c_j$ ($j \in [1, 10]$).

In conventional classification tasks, three evaluation parameters are commonly examined, viz. precision, recall and F-Score. They are calculated as follows:

$$\text{Precision} = \frac{\text{No. of correctly classified documents}}{\text{No. of total classified documents}}$$

$$\text{Recall} = \frac{\text{No. of correctly classified documents}}{\text{No. of total documents in relevant categories}}$$

$$\text{F-Score} = \frac{2pr}{p + r}$$
Table 1. Precision($p$), recall($r$) and $F$-Score values from two text representation models

<table>
<thead>
<tr>
<th>Class label</th>
<th>Graph-Structure Model</th>
<th>Vector Space Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>Environment</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td>Computer</td>
<td>22</td>
<td>18</td>
</tr>
<tr>
<td>Education</td>
<td>21</td>
<td>20</td>
</tr>
<tr>
<td>Transport</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>Economy</td>
<td>21</td>
<td>16</td>
</tr>
<tr>
<td>Military</td>
<td>19</td>
<td>9</td>
</tr>
<tr>
<td>Sports</td>
<td>28</td>
<td>17</td>
</tr>
<tr>
<td>Medicine</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>Art</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>Politics</td>
<td>18</td>
<td>14</td>
</tr>
<tr>
<td>Agriculture</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Law</td>
<td>22</td>
<td>16</td>
</tr>
<tr>
<td>History</td>
<td>23</td>
<td>19</td>
</tr>
<tr>
<td>Philosophy</td>
<td>18</td>
<td>15</td>
</tr>
<tr>
<td>Electronic</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>Avg.</td>
<td>75.43</td>
<td>75.33</td>
</tr>
</tbody>
</table>

Especially, in Formula (7), $p$ and $r$ represent the precision and recall of the classification model respectively.

The experimental process consists of the following three steps: first, extract the feature terms and their corresponding relationships by preprocessing approach; second, represent the selected texts with VSM model and the proposed weighted graph-structure model respectively; third, deploy both the cosine similarity measurement and mcs-based similarity measurement to compute the similarities between texts. In doing so, the similarities of the chosen 300 pieces of texts can be obtained, which can then be used for text classification through Formula (4).

4.3. Experimental results. In the experiments, we randomly choose 15 categories and 20 pieces of texts from each category in different domains (Environment, Computer, Education, Transport, Economy, etc.). Totally 300 pieces of texts are deployed for performance evaluation. Furthermore, the same dataset is analyzed by both graph-structure model and VSM model. After 20-time running, the steady results are gained, as shown in Table 1 and Figure 3. Especially, $F$ denotes the number of texts classified, and $T$ denotes the number of texts correctly classified.

From Table 1, we can find out that the average F-Scores of both VSM model and graph-structure model are almost identical, Figure 3 indicates the slight differences between these two models, in which the VSM model takes a steady effect on different text collections and the graph-structure model holds the high F-Score values in “Education” and “Agriculture” classes but low value in “Military”. To look into this situation, we get back to the dataset and analyze these original corpora classified. It is interesting to find out that the texts in “Education” and “Agriculture” are with large lengths. The average word numbers of these two categories are 7,605 and 6,072 respectively, however, such number in “Military” is only 303 (summarized in Table 2). It implies that the graph-structure model really performs well for the long texts.
To confirm our judgment, we undertake another experiment for those text collections whose lengths are more than 1,000 words. It turns out that we acquire much higher average precision (84.60%), recall (84.29%) and F-Score (0.8428) (shown in Table 1). We contend that the reason behind this situation lies in that the graph-structure model performs well on representing semantic relationships, as well as the structural characteristics of texts. More specially, the longer the text is, the more relationships the graph holds. Therefore, more semantic information of texts can be incorporated for similarity calculation thereby leading to the better classification results. On the other hand, with respect to the VSM model, it depends upon the independent terms extracted from the texts. In this regard, the longer the text is, the larger the number of terms is. Apparently, many terms would affect the classification identification and thus result in a relatively poor classification results.

In order to enable the graph-structure model to be also applicable for the short texts, a modified approach is developed to construct the graph structure. It transforms the “sentence” into “paragraph” for the unit defined in Definition 2.2. In doing so, more mutual information of terms can be retained. Then we repeat the experiment and the results are really inspiring. The average precision increases to 82.53%, recall to 82.33% and F-Score to 0.8234. Obviously, those numbers are superior to the numbers (75.43%, 75.35% and 0.7496 respectively in Table 1) before the modification. Moreover, the F-score values corresponding to the different 15 categories are demonstrated in Figure 4. From Figure 4, it is obvious that the F-Score values after the modification are higher than those values before the modification, especially for the categories “Environment”, “Transport”, “Economy”, “Law”, “Politics”, “Military”, “Agriculture”, “Computer”, “Education”, “Sports”, “Medicine”, “Art”, “Philosophy” and “Electronic”.

Table 2. The average number of words of texts in the given 15 categories

|-----------------|----------------|-----------------|----------------|--------------|
Figure 4. Comparison of pre- vs. after-modified F-Score values

“Military”, “Sports”, “Medicine”, “Art”, “Law” and “Electronic”. Those results indicate that the developed graph-structure model performs well for classification tasks on both large texts and short texts after the modification.

5. Conclusions. In this paper, we present an approach to construct the semantic graph-structure model for Chinese text representation, and develop a mcs-based similarity calculation algorithm for text classification. The merit of the developed model derives from the deployment of weighted edges of the graph, which incorporates more semantically related information in the texts than the unweighted graph-structure models. The primary contribution of this study is to provide a more expressive way to represent the texts, especially for Chinese texts, towards such applications as text classification, text clustering, etc. We have compared our proposed approach with one popular conventional model, i.e., VSM model. The experimental results have indicated that in terms of three evaluation parameters, viz. precision, recall and F-Score, our method clearly outperforms the conventional VSM model on the text classification tasks.

There are still a number of issues to be examined in the future. To begin with, in Section 3.3, we clarify the time complexity of the mcs extraction is $O(|n|^2)$. As the number of nodes in the mcs increases, the $O(|n|^2)$ will become larger. This situation would become worse when very long-length texts are collected for analysis. Therefore, in the future, we will focus on this issue and try to look into the tradeoff between the performance and the execution time. In addition, our proposed semantic-graph-structure model for Chinese text representation opens a door for obtaining more accurate solutions towards the other text mining tasks. Hence, it is substantially significant to apply our presented graph-structure method to the other text mining tasks, e.g., text clustering.

Acknowledgments. This work has been sponsored by the National Natural Science Foundation of China (NSFC) under Grant No. 70771019. It is also partially supported by the National High Technology Research and Development Program of China (No. 2008AA04Z107). The authors would like to thank the editor and the anonymous reviewers for their great help to improve the original manuscript.
REFERENCES


