Automatic Category Generation for Text Documents by Self-Organizing Maps

Hsin-Chang Yang and Chung-Hong Lee
Department of Information Management, Chang Jung University, Tainan, 711, Taiwan
{heyang,leechung}@mail.cju.edu.tw

Abstract

Recently knowledge discovery and data mining in unstructured or semi-structured texts has been attracted lots of attention from both commercial and research fields. The task is not easy to tackle due to the unstructured nature of ordinary text documents. Text data mining approaches has been proposed to resolve such difficulties. One important task for text data mining is automatic text categorization, which assigns a text document to some predefined category according to their correlations. Traditionally these categories as well as the correlations among them are determined by human experts. In this paper, we devised a novel approach to automatically generate categories. The self-organizing map model is used to generate two maps, namely the word cluster map and the document cluster map, in which a neuron represents a cluster of words and documents respectively. Our approach is to analyze the document cluster map to find centroids of some super-clusters. We also devised a method to select the category term from the word cluster map. The hierarchical structure of categories may be generated by recursively applying the same method. Text categorization is the natural consequence of such automatic category generation process.

Keywords: Automatic Category Generation, Text Categorization, Self-Organizing Maps, Text Data Mining

1 Introduction

Text categorization has attracted lots of attention in text data mining fields because we need some 'knowledge' to correctly categorize documents. One of the key knowledge in text categorization process is the selection of categories. Traditionally such knowledge is provided by human experts or some semi-automatic mechanisms which incorporate human knowledge as well as computing techniques, for example, natural language processing. Fully automatic category generation is difficult due to two reasons. First, we need to select some important words that can be assigned as category terms. These words are used to represent categories and provide indexing information for the categorized documents. Such selection was always done by human linguistic experts because we need an insight of the underlying semantic structure of a language, which is hard to automate. Second, the relations among categories, i.e. the structure or hierarchy of categories, are also hard to reveal. A thorough investigation about the semantic relations among category terms must be conducted to establish such structure. Therefore, most of text categorization systems focus on categorizing documents according to some human-specified category terms, rather than generating category terms.

Traditionally, category terms are selected according to the popularity of words in the majority of documents. This can be done by human engineering, statistical training, or a combination of the two. In this work, we provide a method which can automatically generate category terms and establish the structure of categories. We reverse the text categorization process to obtain the category terms. A set of documents are trained by self-organizing maps (SOM) [1] algorithm to generate two maps, namely document cluster map and word cluster map. A neuron in these two maps represents a document cluster and a word cluster respectively. Through self-organizing process the distribution of neurons in the map reveals the similarities among clusters. We select category terms according to such similarities. Dominating neurons in the document cluster map are selected as centroids of some super-clusters, which each represents a general category. The words associated to the same neuron in word cluster map are then used to select category terms. The structure of categories may also be revealed by examining the correlations among neurons in the two maps.
2 Related Works

Text categorization or classification systems usually categorize documents according to some predefined category structure. An example is the work in CMU text learning group [2] where they used Yahoo hierarchy to categorize documents. Another approach is to automatically generate the category terms as well as hierarchy. McCallum and Nigam [3] used a bootstrapping process to generate new terms from a set of human-provided keywords. Rauber and Merkl [4] used the self-organizing map to cluster text documents. They labeled each neuron with a set of keywords that were selected from the input vectors mapped to this neuron. Those keywords that contribute less quantization error were selected. These keywords together form the label of a cluster, rather than a single term for a category. Moreover, the hierarchical structure among these keywords were not revealed.

3 Generating Clusters

3.1 Encoding Documents

Our approach begins with a standard practice in information retrieval (IR) [5] to encode documents with vectors, in which each element of a vector corresponds to a different word. In this work the corpus contains a set of Chinese news documents from CNA(Central News Agency). We extract Chinese words from the documents. A word in Chinese may contain two or more Chinese characters. Binary representation scheme is used to represent the presence of a word in a document. A value of 1 for an element in a vector indicates the presence of the corresponding word in the document.

A problem with this encoding method is that if the vocabulary is very large the dimensionality of the vector is also high. In practice, the resulting dimensionality of the space is often tremendously huge, since the number of dimensions is determined by the number of distinct indexed terms in the corpus. In general, feature spaces on the order of 1000 to 100000 are very common for even reasonably small collections of documents. As a result, techniques for controlling the dimensionality of the vector space are required. Such a problem could be solved, for example, by eliminating some of the most common and some of the rarest words in the preprocessing stage. Several other techniques may also be used to tackle the problem. Examples are multidimensional scaling [6], principal component analysis [7], and latent semantic indexing [8].

3.2 Generating The Word Cluster Map and Document Cluster Map

The word cluster map that is employed for document encoding is produced according to word similarities, measured by the similarity of the co-occurrence of the words. Conceptually related words tend to fall into the same or neighboring map nodes. By means of the SOM algorithm, word clusters can be ordered and organized as nodes in the map. Let \( x_i = \{ x_i_n | 1 \leq n \leq N \}, 1 \leq i \leq M \), be the feature vector of the \( i \)th document in the corpus, where \( N \) is the number of indexed terms and \( M \) is the number of the documents. We used these vectors as the training inputs to the map. The map consists of a regular grid of neurons. Each neuron in the map has \( N \) synapses. Let \( w_j = \{ w_j_n | 1 \leq n \leq N \}, 1 \leq j \leq J \), be the synaptic weight vector of the \( j \)th neuron in the map, where \( J \) is the number of neurons in the map. We trained the map by the SOM algorithm for a time \( T \).

The trained map forms a Document Cluster Map by labeling each neuron with certain documents. The labeling process is described as follows. Every document feature vector \( x_i, 1 \leq i \leq M \) is presented to the map to match a neuron which synaptic weight vector is the closest to \( x_i \). The \( i \)th document is then labeled to the matching neuron. After the labeling process, every document is labeled on some neuron. It is possible that some neurons may not be labeled by any document. The situation occurs when the number of documents are too few compared to the number of neurons or the corpus contains too many conceptually similar document. However, such situation will not diminish the result of the clustering.

In the labeling process those documents with similar keywords will map to the same or neighboring neurons. Since the number of the neurons is much less than the number of the documents in the corpus, multiple documents may map to the same neuron. Thus a neuron forms a document cluster. Besides, neighboring neurons represent document clusters of similar meaning, i.e. high word co-occurrence frequency.

The map forms a Word Cluster Map by labeling each neuron with certain words. We label the neurons by examining their synaptic weight vectors. For the weight vector of the \( j \)th neuron \( w_j \), if its \( n \)th element exceeds a predetermined threshold, the corresponding word of that element is labeled to this neuron. The threshold is a real value near 1. Because we use binary representation for the document feature vectors, ideally the trained map will contain synaptic weights with values near either 0 or 1. Since a value 1 of an element in a document vector represents the presence of a word in that document, an
element with value near 1 in a weight vector also shows that that neuron 'learns' the word. A neuron may be labeled by several words which often co-occurred in the documents. Thus a neuron forms a word cluster. The labeling method may not completely label every word in the corpus. This happens when several neurons compete for a word, result in imperfect convergence of weights. We solve this problem by examining all the neurons in the map and labeling the unlabeled word to the neuron with the largest value of the corresponding element for that word. That is, the nth word is labeled to the jth neuron if \( ||w_{jn}|| = \max_{1 \leq k \leq j} ||w_{kn}||. \)

The word cluster map autonomously clusters words according to their similarity of co-occurrence. Words tend to occur simultaneously in the same document will be mapped to close neurons in the map. For example, the Chinese words for "neural" and "network" often occur simultaneously in a document. They will map to the same neuron, or neighboring neurons, in the map. Words that are not occurred in the same document will map to distant neurons in the map. Thus we can reveal the relationship between two words according to their corresponding neurons in the word cluster map.

4 Automatic Category Generation Process

A neuron in the document cluster map represents a cluster of documents. Documents associated with neighboring neurons contain words that often co-occur in these documents. Thus we may form a super-cluster by combining neighboring neurons. To form a super-cluster, we first define distance between two clusters:

\[
D(i, j) = ||G_i - G_j||, \tag{1}
\]

where \( i \) and \( j \) are the neuron indices of the two clusters and \( G_i \) is the two-dimensional grid location of neuron \( i \). \( ||G_i - G_j|| \) measures the Euclidean distance between the two coordinates \( G_i \) and \( G_j \). We also define the dissimilarity between two clusters:

\[
D(i, j) = ||w_i - w_j||. \tag{2}
\]

We may compute the supporting cluster similarity \( S_i \) for a neuron \( i \) from its neighboring neurons by

\[
S(i, j) = \frac{\text{doc}(i)\text{doc}(j)}{F(D(i, j)D(i, j))}, \quad S_i = \sum_{j \in B_i} S(i, j), \tag{3}
\]

where \( \text{doc}(i) \) is the number of documents associated to neuron \( i \) in the document cluster map and \( B_i \) is the set of neuron index in the neighborhood of neuron \( i \). The function \( F : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \) is a monotonically increasing function. A dominating neuron is the neuron which has locally maximal supporting cluster similarity. We may select dominating neuron by the following algorithm:

1. Find the neuron with the largest supporting cluster similarity. Selecting this neuron as dominating neuron.
2. Eliminate its neighbor neurons so that they will not be considered as dominating neurons.
3. If there is no neuron left or the number of dominating neurons exceeds a predetermined value, stop. Otherwise goto step 1.

A dominating neuron is the centroid of a super-cluster, which contains several clusters. The \( i \)th cluster (neuron) belongs to the \( k \)th super-cluster if

\[
D(i, k) = \min_l D(i, l), \quad l \text{ is a super--cluster}. \tag{4}
\]

A super-cluster may be thought as a category which contains several sub-categories. Let \( C_k \) denote the set of neurons that belong to the \( k \)th super-cluster, or category. The category terms are selected from those words that associated to these neurons in word cluster map. For all neurons \( j \in C_k \), we select the \( n \)th word as the category term if

\[
\sum_{j \in C_k} w_{jn} = \max_{1 \leq n \leq N} \sum_{j \in C_k} w_{jn}. \tag{5}
\]

Eq. 5 selects the word that is most important to a super-cluster since the weight vector of a synaptic in a neuron reflects the willingness that the neuron wants to learn the corresponding input data, i.e. word.
The terms selected by Eq. 5 form the top layer of the category structure. To find the descendants of these terms in the category hierarchy, we may apply the above process to each super-cluster. A set of sub-category will be obtained. These sub-categories form the new super-clusters that are on the second layer of the hierarchy. The category structure can then be revealed by recursively applying the same category generation process to each new-found super-cluster. We decrease the size of neighborhood in selecting dominating neurons when we try to find the sub-categories.

5 Experimental Results

We applied our method on the Chinese news articles posted daily in the web by CNA (Central News Agency). In the preliminary experiments a corpus was constructed by randomly selecting 100 news articles posted in Aug. 1, 2, and 3, 1996. A word extraction process was applied to the corpus to extract Chinese words. Total 1475 words were extracted. To reduce the dimensionality of the feature vectors we discard those words which occur only once in a document. This reduced the number of words to 563. We also constructed a self-organizing map which contains 64 neurons in 8 x 8 grid format. Each neuron contains 563 synapses. The initial training gain was set to 0.4 and the maximal training time was set to 100. Labeling processes were applied to the map to obtain the document cluster map and the word cluster map. The resulting word cluster map and document cluster map are shown in Fig. 1 and Fig. 2, respectively. In Fig. 1 each grid represents a neuron. The neuron indices increase row by row starting with 1 for grid A1. The words labeled on a neuron are listed within the grid. Using the same approach we show the subjects of the documents labeled on the same neuron within a grid in Fig. 2.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Number of Word = 563, Number of Training Vectors = 100
Maximal Training Time = 100, Number of Neurons = 64, Initial Training Gain = 1.0

Figure 1. The word cluster map.

We applied the category generation process to the document cluster map to obtain the category structure. Fig. 3 depicts the result of the first layer. Each circle represents a neuron that is in the corresponding location as in Fig. 1 and 2. The neuron indices are shown at the upper-left corner of each neuron. The size of a circle depicts the supporting similarity of the corresponding neuron obtained by Eq. 3. A dominating neuron for a super-cluster is marked by a cross. The 2-tuple (x, y) beneath each circle shows that the corresponding neuron(cluster) belongs to the super-cluster x where y is its supporting similarity. For example, the 35th neuron has a 2-tuple (1, 3.0) and a cross on it, which shows that this neuron is a dominating neuron for super-cluster 1. The supporting similarity for this neuron is 3.0. Neurons belonging to the same super-cluster
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
<td>サブカテゴリーの配置図</td>
</tr>
<tr>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
<td><img src="image" alt="Figure 2. The document cluster map." /></td>
</tr>
</tbody>
</table>

Figure 3. The supporting similarity for each neuron in the first application of category generation process.

locate closely in Fig. 3. The category terms for super-clusters are shown beneath the 2-tuple. Fig. 4 depicts the final category hierarchy. Each tree depicts a category where the number in the root node depicts the super-cluster found in Fig. 3. Each node in a tree represents a cluster in the document cluster map. The parent node of some child nodes represents a super-cluster. The number enclosed in every leaf node is the neuron index of its associated cluster in the document cluster map. The root node of each tree is the super-cluster found in the first application of category generation process. Text categorization can be achieved by comparing the similarity between the input document and each leaf node in Fig. 4.

6 Conclusions

In this paper, we present a method to automatically generate category terms and structure. The documents were first transformed to a set of feature vectors, in which each component corresponds to a different word and the value of the component reflects the frequency of occurrence of the word in the document. The vectors were used as input to train the
Figure 4. The category structure of the training documents.

self-organizing map. Two maps, namely the word cluster map and the document cluster map, were obtained by labeling the neurons in the map with words and documents respectively. An automatic category generation process was applied to the document cluster map to find some dominating neurons that are centroids of some super-clusters. The category terms of super-clusters were also determined. The same process were applied recursively to each super-clusters to reveal the structure of the categories. Our method used neither human-provided terms nor predefined category structure. Text categorization can easily be achieved in our method.

References


