Chinese text clustering algorithm based k-means

Mingyu Yao¹, Dechang Pi¹, Xiangxiang Cong²

¹ College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu, 210016, PR China
² School of Information Science & Engineering, East China University of Science and Technology
Shanghai, 200237, PR China
my_yao@163.com, dc.pi@nuaa.edu.cn

Abstract

Text clustering is an important means and method in text mining. The process of Chinese text clustering based on k-means was emphasized, we found that new center of a cluster was easily effected by isolated text after some experiments. Average similarity of one cluster was used as a parameter, and multiplied it with a modulus between 0.75 and 1.25 to get the similarity threshold value, the texts whose similarity with original cluster center was greater than or equal to the threshold value were collected as a candidate collection, then updated the cluster center with center of candidate collection. The experiments show that improved method averagely increased purity and F value about 10 percent over the original method.

© 2012 Published by Elsevier B.V. Selection and/or peer review under responsibility of ICMPBE International Committee. Open access under CC BY-NC-ND license.

Keywords: text cluster; k-means; Chinese text

1. Introduction

Text clustering is an important means and methods of text mining, and also a part of data mining. Text clustering is an unsupervised classification of documents, which divides a text collection into several subsets called clusters, the text of each cluster has greater similarity than the one in different cluster. Text clustering is the process of classing text automatically by data mining, machine learning and neural network technology etc.

Text description model is usually using VSM(Vector Space Model) in text clustering[1]. Steinbach and others compared applicability of method based on hierarchical and method based on classification in text clustering[2][3], thought that k-means and bisecting k-means algorithm not only could get good clustering results but also cost time with a liner relationship between the amount of text, so it is good for large scale text
clustering. Dhillod and others said that, similarity measure using cosine would get significantly better clustering result than using Euclidean distance text clustering[4].

The main application of text clustering currently are: Can serve as a pre-processing step for SDS (Multiple Document Summarization) and other NLP (Natural Language Processing) applications, such as the multi-document summarization system Newsblaster developed by Columbia University [5]; Clustering the results returned by search engine, allowing users to quickly locate the information they need [6]; Can be used to optimize the results of text classification, such as work of Y.C.Fang, S. Parthasarathy, F. Schwartz and others from Ohio State University [7]; Automatically organize the document collection, such as Scatter / Gather [8] is a document browser system based on clustering.

The remaining part of the main contents of this article is: Section 2 analyzed Chinese text clustering algorithm base k-means detailed and improved it; Section 3 compared improved method by experimental results; Section 4 concluded the paper and pointed out the next job.

2. **Improve Chinese text clustering algorithm base k-means**

2.1 text clustering

Text clustering process generally includes the following steps:

1. Model description: Including feature extraction and selection, demonstrating the data object into a form which is suitable to calculate in algorithm.
2. Define the distance formula between the measurement models.
3. Do clustering algorithm experiments.
4. Evaluate the output.

Specific process is shown in Figure 1:
2.2 Carving Chinese Text

Text is posed by strings which are posed by large number of characters, it is a form of unstructured data. In order to turn it into forms can be handled by computer, pretreatment is necessary. The first preprocessing key step is curving words, which separates continuous sequence of words into scattered independent meaning word sequence by a certain rule, and then filter out stop words of them and get a collection of text key words, these words constitute the initial characteristics of items (words) collection.

English words separated by spaces between, it is relatively simple in terms of semantic accuracy and technical complexity; Chinese text is composed of continuous words, only separated by sentences and paragraphs, but the words should be the smallest unit during text message processing. Thus, it is more complex and difficult than Western text language. In this paper, we curving text using ICTCLAS2009 system developed by Institute of Computer Technology which tags word nature and gives word weight for curved words, it provides a convenient for text vector description.
2.3 Text Demonstration

In this paper, we use fluent vector space mode VSM[1], whose main idea is mapping each document to a point of vector space composed by a group of normalized orthogonal eigenvector. VSM grants that: document category is only decided by words or vocabulary and their frequency appeared in it, but has nothing to do with the appeared position or order. Word is usually selected as feature item in VSM. For a document \( d_i \), \( t_j \) \((j=1, \cdots, n)\) is a word different from each other, weight of feature item \( t_j \) in document \( d_i \) is marked as \( w_{ij} \). document \( d_i \) is demonstrated as formula (1):

\[
V \left( d_i \right) = \{(t_j, w_{ij})\}_{i=1}^n
\]  

2.4 TF-IDF Model

We use curving text program to cut text into word sequence and pre-process it, including removing stop words, punctuation and numbers. Then extract all appeared words in document to be TextVector, count the frequency \( TF(i, j) \) of each word in TextVector appeared in document, where \( i \) refers the \( i \)-th word in TextVector, \( j \) refers the \( j \)-th document.

Regulating TF, Calculated by formula (2):

\[
TF_{reg}(i, j) = \frac{TF(i, j)}{\max_{k \in TextVector}(TF(k, j))}
\]  

Calculating IDF by formula (3):

\[
IDF(i) = \log \frac{N}{N_i}
\]  

Where \( N \) is the total number of documents, \( N_i \) refers the times \( i \)-th word of TextVector appeared in how many documents.

Method for calculating the eigenvector is formula (4):

\[
TextVector(i, j) = TF_{reg}(i, j) * IDF(i)
\]  

Suppose feature vectors values of two documents \( d_x \), \( d_y \) are:

\[
(x_i)_1^n \text{ and } (y_i)_1^n, \text{ where } x_i, y_i \in \text{real}; \text{ Two document similarity is calculated as formula (5):}
\]

\[
sim(d_x, d_y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}
\]

2.5 Improved k-means algorithm

Basic k-means algorithm is relatively simple. First, select \( k \) initial centers, where \( k \) is a user-specified parameter, that is the desired number of clusters. Assign each point to the nearest center, points who are assigned to the same center is a cluster. Then update the center of each cluster according to
assigned point. Repeat assignment and update steps until the cluster does not change, or equivalently, until the center does not change. Algorithm described in Table 1.

Table 1. Description of k-means steps

<table>
<thead>
<tr>
<th>Basic k-means algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Select k-point as the initial center</td>
</tr>
<tr>
<td>2: repeat</td>
</tr>
<tr>
<td>3: Assign each point to the nearest center to form k cluster</td>
</tr>
<tr>
<td>4: Recalculate the center of each cluster</td>
</tr>
<tr>
<td>5: until center does not change</td>
</tr>
</tbody>
</table>

In text clustering experiments using k-means algorithm, we found that apart from the initial cluster centers, isolated point which is not similar document, also influence clustering results heavily. This makes clustering results very unstable, so this paper presents an improved method, whose idea is selecting documents more similar to the original cluster center to calculate average value as new center when updating cluster center. Improved k-means algorithm described in Table 2 below:

Table 2. Improved k-means algorithm

<table>
<thead>
<tr>
<th>Improved k-means Chinese text clustering algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Select k points {c_1, c_2, ..., c_k} as initial center</td>
</tr>
<tr>
<td>2: Calculate similarity between each document d_i and each c_j, classify most similar cluster to get {C_1, C_2, ..., C_k}, and record the similarity</td>
</tr>
<tr>
<td>3: Calculate the average similarity for each cluster C_i, marked as meansSim</td>
</tr>
<tr>
<td>4: Select documents set {d_1, d_2, ..., d_m} from C_i whose similarity between c_i are greater than ((1 + \theta)*\text{meansSim}, \theta \in [-0.25, 0.25])</td>
</tr>
<tr>
<td>5: Calculate average point of {d_1, d_2, ..., d_m} as new center for C_i</td>
</tr>
<tr>
<td>6: repeat step 2 to step 5 until Center does not change</td>
</tr>
</tbody>
</table>

Improved k-means clustering algorithm's time complexity analysis:

Let n be the size of the text set, k is the number of clusters, t is the number of iterations. Compared to not improved k-means, calculating each cluster similarity is added, whose time-consuming is \(O(nkt)\), the total time complexity is also \(O(nkt)\), so efficiency is not affected over the earth while clustering results are improved.

3. Analysis of experimental results

This paper uses laboratory Sogou Sohu R & D Center of Corpus text classification corpus: http://www.sogou.com/labs/dl/c.html. Text classification corpus saved from Sohu news site after a large number of hand-sorting and classification of news corpus and the corresponding classification information. Experimental hardware environment: AMD Ath.64 QL65 2.10GHz, 2GB memory; Software environment: Windows 7 OS, VS2005 IDE; Programming Language: C++.
Experiment selected some corpus from Sogou corpus, including 8 species: “C000008 Finance, C000010 IT, C000013 Health, C000014 Sports, C000016 Tourism, C000020 Education, C000022 Recruitment and C000023 Culture“, 30 documents each type, 240 documents in total, about 350 words per document.

To test the effect of the improved algorithm, this paper selected a sample test results, compared the results of purity and F-Measure values of two methods, as shown in Figure 2 and Figure 3 below:

![Figure 2. Compare to purity of clustering result](image1)

![Figure 3. Compare to F value of clustering result](image2)

Figure 2 shows that the improved k-means algorithm improved the purity of some clustering results relatively. Figure 3 shows the F value has risen markedly, that the clustering accuracy and stability are improved.

4. **Summary and Outlook**

This paper analyzes the process of Chinese text mining based on k-means algorithm. We presented an improved k-means clustering method based on isolated point problem found in experiments, and proved that the algorithm is correct and effective by experiments. Although k-means clustering results than have improved, but the overall result is not very satisfactory, The reason is the different characteristics of the meaning of the word is assumed to be different, and this is precisely an important factor leads to not ideal...
results. The next step is to study combines the concept of [9] of the text clustering, further enhance the Chinese text clustering results.

References