A Clustering Algorithm of No-Word-Segmentation for Chinese Search Engine Results

Deqing Wang  Hui Zhang  Liping Zhao  Ke Xie
(Beihang University, State Key Lab of Software Development Environment, Beijing 100083)
wangdeq@nlsde.buaa.edu.cn, hzhang@nlsde.buaa.edu.cn

Abstract

Along with information on the Internet increasing dramatically, People usually search and locate information that they needed by search engines. Clustering search engine results is an effective method to help people select information needed from the list of search engine results. The paper presents a clustering algorithm of no-word-segmentation for Chinese search engine results (CANWS). The algorithm firstly preprocesses the search engine results and then computes the similarities of the results based on the same sub-string. Lastly it clusters the results based on the similarity matrix. The paper also gives test and analysis of the algorithm performance by experiments.

Keywords: search engine results; clustering results; similarity algorithm; clustering algorithm

1. Introduction

Web documents’ clustering is an unsupervised document classification method. Its intention is to divide a set of documents into several subsets, with which the similarities of documents within a subset should be bigger as possible and the similarities between subsets should be smaller as possible. Now there are three ways on web documents clustering: Linked-based Clustering [1], clustering web search results based on documents clustering [2] and based on Users’ Feedback and Expert’s Agreement [3]. Now there are many text clustering algorithms, such as K-means [4], HAC [5-7] STC [8] and so on. The paper presents a new clustering algorithm of no-word-segmentation for Chinese search engine results.

2. Disadvantages of traditional clustering algorithms

The traditional document clustering algorithms such as K-means, HAC are based on Vector Space Models [9] (VSM), which cluster the documents according to the similarity of vectors which adopt the cosine angle between the vectors to represent.

However, VSM has a considerable shortage. Firstly, VSM needs segmenting Chinese texts, so the precise of the segmentation algorithm directly influences the clustering results. Secondly, dimension disaster [10]. Even for middle-scale documents collection, the number of words is up to several tens of thousands or even hundreds of thousands ones. From experiments, dimension of features is up to 7,286 when the number of documents is only 200. (Shown in Figure 1)

Figure 1: the relation between feature dimension and the number of documents

There are a lot of methods to reduce feature dimension. But the main idea is to save some important features and delete useless ones, which brings the loss of effective information, affecting the clustering results.

3. CANWS system workflow

The CANWS, different from the traditional clustering algorithm, doesn’t need word segmentation of Chinese texts, so the CANWS system doesn’t contain the module of Chinese words segmentation. The workflow of the system is shown in Figure 2.

Figure 2: Workflow of the CANWS System
Preprocesses to Chinese text is to delete the non-Chinese words in the search engine results. Then it computes the similarities between results. Clustering module is to cluster the search engine results based on the similarity matrix. Description Module of clustering results is to generate a meaningful label which can summarize the contents of each cluster. The paper will introduce preprocess of search engine results, the computation of similarities and the clustering algorithm (CANWS).

4. Preprocess

In order to improve the coverage of information, major search engines (such as Google) return many non-Chinese search results. Besides, there are some English characters, spaces, punctuation marks in Chinese search engine results. The paper mainly does research for Chinese search engine results, so it will affect the clustering results if we don’t delete them. In order to obtain better results, Major operations in preprocesses are as following.

1. Delete non-Chinese characters.
2. Delete spaces and punctuation marks.

Then we get the search engine results which only contain Chinese characters.

5. Algorithm for similarity

5.1 Edit Distance

We distinguish string s1 from s2 through calculating the minimum edit distance \[d_{Levenshtein}(s1, s2)\] (Levenshtein Distance). The so-called edit distance means: the minimum times of operations to let string s1 and string s2 into the same string. It is first presented by a Russian scientist named Levenshtein.

The operations contain,
1. Change character ch1 to ch2.
2. Delete a character.
3. Insert a character.

For example, s1="北京航空航天大学" s2="北京航空航天大". s1 will be the same as s2 if we insert a character (ch="学") into s2. That is, \(d(s1, s2) = 1\).

Edit distance is usually used for fast approximate string matching between sentences. Generally speaking, the greater the distance, the lower the sentence similarity; the smaller the distance, the greater the sentence similarity. But we must define certain algorithm of transformation from edit distance to similarity. There are two reasons. Firstly, edit distance could not stand for the similarity between sentences, For example, sentence A: “北航计算机学院软件开发环境国家重点实验室”, sentence B: “北航软件开发环境国家重点实验室”, \(d(A, B) = 5\). sentence C: “北京大学”, sentence D: “清华大学”, \(d(C, D) = 2 < d(A, B)\).

However, the similarity between A and B is greater than that between C and D. Secondly, The operations to calculate edit distance are not flexible and the minimum operating unit is character, which doesn’t adapt to Chinese text.

5.2 Calculating Formula for Similarity

The new calculation method of similarity just adapts to Chinese search engine results. We think the similarity between sentence A and B is decided by the same substrings.

That is, a) the more the same substrings, the greater similarity. b) the longer the same substring, the greater similarity. Supposed, string A and string B contain substrings C, D and E. And length of C=length of D+ length of E. In this case, the similarity of C > similarity of D + similarity of E.

We use maximum matching algorithm to search the same substrings (In the paper, the length of substring \(>=2\)).

Some definitions are as following,

a) String A and string B have the sequences of the same substrings \((S_1, S_2, S_3, \ldots S_{n-1}, S_n)\), \(S_i\) is i-th substring between A and B.
b) The length sequence of the substring: \((\text{length}_{(S_1)}, \text{length}_{(S_2)}, \ldots \text{length}_{(S_{i-1})}, \text{length}_{(S_i)})\), \(\text{length}_{(S_i)}\) stands for the length of the substring \(S_i\).
c) The frequency of substrings in sentence A: \((\text{times}_{(S_1, A)}, \text{times}_{(S_2, A)}, \ldots \text{times}_{(S_{i-1}, A)}, \text{times}_{(S_i, A)})\), \(\text{times}_{(i, A)}\) stands for the frequency of the substring \(S_i\) in sentence A. And the frequency in sentence B: \((\text{times}_{(S_1, B)}, \text{times}_{(S_2, B)}, \ldots \text{times}_{(S_{i-1}, B)}, \text{times}_{(S_i, B)})\), \(\text{times}_{(i, B)}\) stands for the frequency of the substring \(S_i\) in sentence B.
d) \(\text{length}_{(A)}\) stands for the length of sentence A and \(\text{length}_{(B)}\) stands for the length of sentence B.

The calculating formulas are as following:
\[
\text{Sim}_{(A,B)} = \frac{\text{Imp}_{(A,B)} + \text{Imp}_{(B,A)}}{2} (\text{Formula 1})
\]

\[
\text{Imp}_{(A,B)} = \sum_{i=1}^{n} \left( \text{times}_{(i)}^{(A)} \times \left( \frac{\text{length}_{(S_i)}}{\text{length}_{(A)}} \right)^\alpha \right)
\]

\[
\text{Imp}_{(B,A)} = \sum_{i=1}^{n} \left( \text{times}_{(i)}^{(B)} \times \left( \frac{\text{length}_{(S_i)}}{\text{length}_{(B)}} \right)^\alpha \right)
\]

\[
\text{Parameter } \alpha \text{ is a constant. } \alpha \in [1, 2]
\]

**6. CANWS**

According to the above calculating method of similarity, the similarity matrix for documents is obtained (shown in Table 1)

<table>
<thead>
<tr>
<th></th>
<th>R_1</th>
<th>R_2</th>
<th>R_3</th>
<th>...</th>
<th>R_N</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_1</td>
<td>Simi_12</td>
<td>Simi_13</td>
<td>...</td>
<td>Simi_1N</td>
<td></td>
</tr>
<tr>
<td>R_2</td>
<td>Simi_23</td>
<td>...</td>
<td>Simi_2N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R_3</td>
<td>...</td>
<td>Simi_3N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R_N</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 similarity matrix of documents

In table 1, Simi_{ij} (1 \leq i, j \leq N) is the similarity between R_i and R_j.

CANWS is as following,

1. if all documents are clustered, finished. if there are some documents which are still not clustered, find two documents with the biggest similarity to form a cluster C, turn to 2;

2. add the document D meeting condition A to cluster C from left documents, update the threshold F. if there no documents meeting the condition A, add the cluster C to the collection of clusters and remove the documents in cluster C from uncluttered documents. turn to 1.

**Condition A:** the similarity between document D and each documents in cluster C is bigger than threshold F.

\[
F = Value \times \beta = \frac{1}{n} \sum_{j=i+1}^{n} \sum_{i=1}^{n} Value_{ij} \times \beta,
\]

\[
\overline{Value} = \frac{1}{n} \sum_{j=i+1}^{n} \sum_{i=1}^{n} Value_{ij} \text{ is the average of similarity in one cluster. } Value_{ij} \text{ is the similarity between the result item Ri and Rj. } \beta \text{ is a constant, and } \beta < 1.
\]

**7. Algorithm Analysis**

**7.1 Analysis of Parameter**

In CANWS, the similarity calculating formula becomes a key factor. From the above analysis it is known that the longer the same substring, the greater the similarity.

In formula 2 and formula 3, \(\alpha\) is a constant and \(\alpha \in [1, 2]\), the final value is obtained from the following experiment.

The experiment is to analyze one hundred search queries. \(\alpha\) is increased by degrees 0.1 from 1 to 2. The statistic data is as shown in Figure 3. (x-axis is the number of clustering results; y-axis is the number of queries.) The expected number of clustering results is between 10 and 20.

From the figure, it is obvious when \(\alpha = 1.4\) (the black curve) the number of clustering results is mostly in [10, 20].

**7.2 Analysis of Results Clustering**

In this section, we evaluate the clustering results with vivisimod\(^\text{14}\) (a famous meta-search engine, which obtains “Best Meta-Search Engine” appraised by Search Engine Watch in 2002 and 2003). We just select and cluster the Chinese search engine results. The clustering results is as shown in Table 2 (keywords="科技创新").
Table 2 the clustering results between vivisimo and CANWS

<table>
<thead>
<tr>
<th>vivisimo</th>
<th>CANWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>以诚为本 团结拼搏</td>
<td>以诚为本 团结拼搏</td>
</tr>
<tr>
<td>清源电动车</td>
<td>清源电动车</td>
</tr>
<tr>
<td>第三届中国青少年科技创新奖颁发</td>
<td>第三届中国青少年科技创新奖颁发</td>
</tr>
<tr>
<td>无</td>
<td>表彰科技创新先进集体和个人</td>
</tr>
<tr>
<td>无</td>
<td>无</td>
</tr>
<tr>
<td>体验科学 健康成长</td>
<td>无</td>
</tr>
</tbody>
</table>

From Table 2, it is known that the clustering results appeared in vivisimo mostly appears in CANWS. But some meaningful clusters (for example, “科技创新体系”) do not appear in vivisimo. CANWS can get better Chinese clustering results from many experiments on different keywords.

Besides, from experiments it is known that the distribution interval of CANWS’s clustering number is [10, 20]. But STC [8] algorithm’s distribution interval is [15, 25], which is shown in Figure 4 (x-axis and y-axis are the same as Figure 3). So CANWS can get better Chinese results clustering number.

From Figure 5, it shows that the efficiency of CANWS is higher than STC [8] with Chinese segmentation. x-axis is the number of documents and y-axis is the average time for clustering these documents. The number of Chinese search engine results increases from 50 to 100.

At the same time CANWS does not need segmenting words, so it has higher efficiency than STC.

8. Conclusion

From experiments, it is known that CANWS has some improvements in clustering effect, efficiency or the number of clustering results compared to the well known clustering systems (vivisimo) or algorithms (K-means, STC).

However the CANWS system still has some disadvantages. For example, the description labels of clusters are not good. Especially, some labels are not meaningful (“科技创新体系”). These problems will be solved in future research.

References

[2] Hua-Jun Zeng Qi-Cai He Zheng Chen Wei-Ying Ma Jinwen Ma, Learning to Cluster Web Search Results