Abstract

The FRDC team participated in the Chinese task of the NTCIR-11, including subtopic mining and document ranking sub-tasks for Chinese language, respectively. In this paper, we propose two methods to build the two-level hierarchy subtopics. Our methods gain high F-score and h mean score respectively. In the document ranking subtask, we adopt various features for relevant webpage retrieval and document ranking.

Subtopic Mining Methods

- **Document Clustering Method**
  - Cluster the candidate queries to get the second-level subtopics, and then generate the first-level subtopics basing on the second-level subtopics.

- **Query expansion.**
  - Convert training document into word vector. After vector segmentation, the document can be changed into word vector pre-processing. TF-IDF is adopted as the weight scheme, in this way, the query is presented by the word vector from its training document.

- **Initial clustering.** For all the queries we firstly use document clustering to generate the second-level subtopics. In order to find the optimal α, we set it from 2 to 10 and then select the best result through the following methods.

- **Select optimal clustering method.** The optimal clustering result is decided by an inner distance dist, score described as:

  \[ \text{dist} = \text{min}(\text{score}(\alpha)) \]

- **Sort second-level subtopics.** Suppose \( q_i \) is the optimal clustering result, \( q_j \) is the topic word list of \( c_i \) obtained by LDA model, \( q_k \) is the ith second-level entry in \( c_i \). We use cosine similarity for \( q_j \) and \( q_k \) in order to find the best result.

- **Generate first-level subtopic.** We adopt some unsupervised technologies to extract meaningful phrases from the training documents as the first-level subtopics, such as Accessor Variety and C-value.

\[ \log(\text{score}(\alpha)) = \frac{\sum_{i=1}^{n} \log(\text{size}(\text{topic}_i))}{\sum_{i=1}^{n} \text{size}(\text{topic}_i)} \]

- **Vectorize the training documents.** This step is almost the same as the DC model. The only difference is using LDA model to obtain topic word as the word vector.

- **Classification.** BaiduPedia entries are considered as the first-level subtopics and the entry pages are exploited as the knowledge base. We just classify all the candidate queries according to the knowledge base. The classification process is based on the rules below:

  1. For a topic with just 1 BaiduPedia entry, we take this BaiduPedia entry as the first-level subtopic of this cluster.
  2. If the weight of a query item in this BaiduPedia entry is 1, then we set the same weight of this query item in the whole cluster.

- **Merge the classification and the clustering results.**
  - If half or more than half query items in this cluster belong to this BaiduPedia in the classification result, we judge that this cluster is related to this BaiduPedia entry, and take this BaiduPedia entry as the first-level subtopic of this cluster.
  - If less than half query items in this cluster belong to this BaiduPedia entry, we judge that this cluster is not related to this BaiduPedia entry, and we extract the frequent key words as the candidate name of this cluster.

- **Subtopic ranking.** We first calculate the ranking score of the second-level subtopics. Since all the second-level subtopics are real queries, we can easily get the number of web search engine results for each subtopic, and this is the only factor for ranking.

\[ \text{rank}_{\text{second level}} = \frac{\log_e(\text{num of results})}{\log_e(10)} \]

- Then we calculate the weight of first-level subtopic with the sum of its two second-level subtopics’ weight value, and process a normalizing step for the first-level subtopics.

Document Ranking Methods

In the document ranking task, we exploit various features for relevant page retrieval and ranking. Our method is described below.

- **Data preparation.** We extract title, anchor, body parts from the webpage.

- **Query expansion.** We try to extract such key words that are related to the given query. For example, when retrieving query “http://,” we may input more key words such as “computer game” so as to find more relevant webpages. We extract key words in several ways:
  - Run LDA model on the training documents of the candidate queries to obtain the topic word list as key words.
  - Extract high-frequency segmentations from training documents as key words and eliminates single character segmentations.
  - Extract segmentations with high TF-IDF value as key words. Combine single character segmentations with other key words generated together to build new key words.
  - According to the two-level hierarchy structure of SIR result, for the query below a subtopic that has been extracted, add its keywords to the subtopic.

- **Feature selection for document ranking.** We exploit various features for ranking the webpage. The features are described as follows:
  - Query coverage. The segmentation number of a query that one webpage covers in the title/anchor/body parts.
  - Keywords coverage. This value is calculated in the title/anchor/body parts of the webpages. If a query exists in one of the parts, its value equals 1 only when the query exists in all of the parts, otherwise the value is 0.
  - TF-IDF similarity. Calculate the cosine similarity between the body of candidate webpage and the training document of the query.
  - Keywords weight of the body part in a webpage.

\[ \text{KeyWeight} = \sum_{i=1}^{\text{num of keywords}} \left( \frac{\text{tf}(i) \cdot \text{idf}(i)}{\text{num of keywords}} \right) \]

- **Webpage classification.** In order to judge whether a webpage is related to a query or not, we manually generate 100 query-document pairs for training a classifier. The output of the classifier are not related, marginally related and related. We use the SVM classifier in our experiment.

- **Step 5. Document ranking.** We calculate and normalized the above features, then the sorted result is generated.

Conclusion & Future work

In the subtopic mining task, our two methods achieve high F-score and h mean score respectively. However, our topic ranking method is not yet mature. This results in low F-score for both DC and CC method. We will further improve the expansion and ranking system in order to make it more useful in the next expansion and it limited the coverage of our query candidates. We tried to merge the results of DC and CC method, but it didn’t yield good results.

In the document ranking task, we exploit various features for relevant webpage retrieval and document ranking.

In the future, we need to improve:
  - More data bases to expand query candidates, such as Wikipedia.
  - Improve the query sorting method.
  - Improve the ranking system.

Table 1. Data Set for Subtopic Mining(Skip) & Document Ranking(DR)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Category</th>
<th>Length</th>
<th>Score</th>
<th>DC Score</th>
<th>DR Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>3.960 / 1.00</td>
<td>1.400 / 1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2A</td>
<td>3.323 / 1.00</td>
<td>1.400 / 1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td>3.000 / 1.00</td>
<td>1.400 / 1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Information Technology Laboratory, Fujitsu Research & Development Center Co., Ltd, Beijing China