FRDC at the NTCIR-11 IMine Task

Zhongguang Zheng, Shuangyong Song, Yao Meng, Jun Sun

{zhengzhg, shuangyong.song, mengyao, sunjun}@cn.fujitsu.com

Abstract

The FRDC team participated in the IMine task of the NTCIR-11, including subtopic mining and document ranking sub-tasks for Chinese language. In the subtopic mining subtask,

we propose two methods to build the two-level hierarchy subtopics. Our methods gain high F-score and H-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 results.

- Convert training document into word vector. After word segmentation, the document can be changed into word vector presentation. TF-IDF is adopted as the weight scheme. In this way, the one 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 *k*, we set k from 2 to 10 and then select the best result through the following methods.
- Initial clustering. For all the queries we firstly use document clustering to generate the second level subtopics. In order to find the optimal *k*, we set k from 2 to 10 and then select the best result.
- Select optimal clustering result. The optimal clustering result is decided by an inner distance $dist_{inner}$ score described as:

$$dist_{inner} = \frac{\sum_{d_i,d_j \in c_r} sim_{topiclist}(t_i,t_j)}{|c_r|} \quad (1)$$

- Sort second level subtopics. Suppose c_r is the optimal clustering result. t_c is the topic word list of c_r obtained by LDA model. d_i is the *ith* document in c_r. wv_i is the word vector presentation of d_i. The second level subtopics are sorted according to the cosine similarity between wv_i and t_c.
- 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.

$$\mathrm{AV}(s) = \min\{L_{av}(s), R_{av}(s)\} \quad (2)$$

$$C-value = \begin{cases} \log_2|a| \cdot f(a) & a \text{ is not nested} \\ \log_2|a| \left(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right) & otherwise \end{cases} \quad (3)$$

- where *a* is the candidate phrase, *f(.)* is the frequency of *a*. T_a denotes the phrase set that contains *a* as substring. $P(T_a)$ is the size of T_a. We select the phrase with the highest *AV* and *C-value* scores. In this way, we obtain the two-level hierarchy subtopics.

■ Classification & Clustering (CC) Method

BaiduPedia is exploited as an external knowledge base in this method. We consider BaiduPedia entries of the test set as the first level subtopics.

- Vectorize the training documents. This step is almost the same as the DC method. The only difference is using LDA model to obtain topic words 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.
  - For a topic with more than 1 Baidupedia entries, we take each Baidupedia entry as one class, and 1NN method with Euclidean distance is used to classify the second level candidate subtopics about this topic.
  - For a topic with just 1 Baidupedia entry or with no Baidupedia entry, we take all the candidates as one class.
- Document clustering. A threshold-based clustering method is designed to cluster candidate second level subtopics. We randomly sample some "candidate couple", and the threshold in the clustering method is calculated as the average value of the Euclidean distance between two candidates in each "candidate couple".

$$s_n = \left\lfloor \frac{c_n(c_n-1)}{2} / 100 \right\rfloor \quad (4)$$

- 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 in the classification result, we judge that this cluster is not related to this BaiduPedia entry, and we extract frequent keywords as the first level subtopic 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.

$$rs = \frac{\log(pn_{st}+1)}{\log(\max\_pn_{st}+1)} \quad (5)$$

- Then we calculate the weight value of a first level subtopic with the sum of its related 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. Thus we want to extract some key words that are related to the given query. For example, when retrieving query "先知 (prophet) dota", we want to input more key words such as "computer game" so as to find more relevant webpage. 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 SM result, for the query below a subtopic that keywords 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 below:
  - 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 parts of one webpage. *w* is the title of the webpage, *K(x,y)* equals to *1* only when keyword *i* exists in *w*, operator $|\overline{w}|$ calculates the length of the string except keywords. When the number of keyword in the string less than two, the value is zero.

$$KeyCoverage = \frac{\sum_{i=0}^{N} K(w, key_i)}{|key|\log(|\overline{w}|)} \quad (6)$$

  - 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.

$$KeywordWeight = \sum_{i=0}^{N} \sum \{C(w, key_i) * TFIDF(key_i) * \alpha(t)\} \quad (7)$$

  - Features inspired by Microsoft Research. We calculate the sum of term frequency, min of term frequency, max of term frequency, mean of term frequency and variance of term frequency.
- Webpage classification. In order to judge whether a webpage is related to a query or not, we manually labeled 100 query-document pairs for training a classier. The output of the classier are not related, marginally related and related. We use the SVM classier in our experiment.
- Step 5. Document ranking. We calculate and normalized the above features, then the sorted result is generated.

Results

■ Data Set for Subtopic Minging(SM) & Document Ranking(DR)

Resource	Official	Size	Training Document	Task
Test Set	Yes	50	----	SM
QuerySuggestion	Yes	1853	Google	SM/DR
RelatedQueries	Yes	2321	Google	SM/DR
BaiduPedia Entries	No	117	BaiduPedia	SM
SogouT(Ver. 2008)	Yes			DR

■ Subtopic Mining

Runs	H-Score	F-Score	S-Score	H-Measure	Method
5A	0.5377 5/19	0.5004 11/19	0.3139 12/19	0.1757 8/19	CC
4A	0.5436 3/19	0.4782 14/19	0.2715 14/19	0.1724 11/19	CC
1A	0.2931 17/19	0.7191 1/19	0.3110 13/19	0.1327 15/19	DC
3A	0.2897 18/19	0.7191 1/19	0.3214 11/19	0.1326 16/19	DC
2A	0.3257 16/19	0.5045 9/19	0.2381 15/19	0.1032 19/19	CC

■ Document Ranking

Runs	Coarse-grain results	Fine-grain results	Method
1A	0.4619 4/10	0.4118 4/10	DC
3A	0.4440 5/10	0.3950 5/10	DC
2A	0.3899 6/10	0.3402 6/10	CC
5A	0.3841 7/10	0.3338 7/10	CC
4A	0.3746 8/10	0.3240 8/10	CC

Conclusion & Future work

- In the subtopic mining task, our two methods achieve high F-score and H-score respectively. However, our topic ranking method is not yet mature. This results in low S-Score for both DC and CC method. Some useful external knowledge base was not used for query 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:
  - Use more data bases to expand query candidates, such as Wikipedia.
  - Improve the query sorting method.
  - Find ways to merge DC and CC results.

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