Introduction

This paper describes the key points of the experiments we conducted with the NTCIR-8 IR4QA system. The basic assumption we based on is that a document is more relevant to a query if it is supported by multi-evidence returned from different methods. So we create three indexes of different formats and then retrieve them to examine the documents returned. If a document appeared more than once, it will be ranked higher than the others, although, as we will see later, the rule is not just so simple since we will assign weights to different indexes. To avoid query drift, we use external resources to perform query expansion. We conduct our experiments by using the Lemur Toolkit (http://www.lemurproject.org) and the official evaluation results show that our method can do improve the performance of the system.

QUERY PROCESSING

We do query processing first since we need to retrieve from different index format.

TERMINAL

QUESTION CLASSIFICATION

For different question types, we will do specific query expansion later. So we must classify the questions into different types now. We classify the questions into 3 types based on the occurrence of the key-strings together with some conflict rules. The key-strings are extracted from 890 questions from Sina iAsk (http://iask.com) and Baidu Zhidao (http://zhidao.baidu.com/) with statistical approach.

PERFORMANCE COMPARISON ON WEIGHTING

We perform query expansion according to the question types mentioned earlier. For questions belong to BIOGRAPHY and DEFINITION, we use Open Category in Baidu Baike (http://baike.baidu.com/) to do the expansion. Otherwise, we adopt the Related Searches provide by Wanfang (http://www.wanfangdata.com.cn) as evidences.

QUERY EXPANSION

Our basic idea is that documents supported by more evidences are more relevant to the query and we considered the results retrieved from different indexes as the evidences. Lemur Toolkit was adopted to create three indexes of different types. KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index. After retrieving, we re-rank the result documents by a simple interpolating algorithm.

WEIGHTING

The importance of terms in a query is not the same. Term weight reflects the discriminative power of the term in query. A proper weighting scheme could enhance the retrieved evidences and document. However, if the system does not support weighting operator, there’s no weighting in the process of KeyFile-Unigram-Index and KeyFile-Word-Index generators. To examine the term weight in Indri-Word-Query, we design an experiment which assigns weights to terms relying on how useful they are likely to be in determining the relevance of a document. We use the following scheme for term weighting:

1. For questions of BIOGRAPHY and DEFINITION, there is no term weighting.

2. For questions of RELATIONSHIP, we assign weights to terms according to the collection frequency of two objects in the question. When the frequencies of the two-objects are not in the same scale, we increase the weight of low frequency object. Otherwise, there is no weighting.

3. For question of other types, we increase the weights of the most important two key words, which are judged by Hallang API (www.hallang.com).

Performance comparison on weighting in the experiments with the NTCIR-7 IR4QA CS test collection

Re-ranking algorithm

Description: This algorithm is designed to implement the document ranking task according to the returned document lists from three indexes mentioned above. For each returned document list from above three indexes, we adopt the Related Searches provide by Wanfang to solve the vocabulary mismatch. Our experiments show that combination of indexes using different index units helps the retrieval performance. Word using combination could obtain a more satisfactory result in general.

QUERY EXPANSION

We perform the query expansion according to the question types mentioned earlier. For questions belong to BIOGRAPHY and DEFINITION, we use Open Category in Baidu Baike (http://baike.baidu.com/) to do the expansion. Otherwise, we adopt the Related Searches provide by Wanfang (http://www.wanfangdata.com.cn) as evidences.

Performance comparison on weighting in the experiments with the NTCIR-7 IR4QA CS test collection

Re-ranking algorithm

Description: This algorithm is designed to implement the document ranking task according to the returned document lists from three indexes mentioned above. For each returned document list from above three indexes, we adopt the Related Searches provide by Wanfang to solve the vocabulary mismatch. Our experiments show that combination of indexes using different index units helps the retrieval performance. Word using combination could obtain a more satisfactory result in general.

RE-RANK

For each returned document list from above three indexes, we adopt the Related Searches provide by Wanfang to solve the vocabulary mismatch. Our experiments show that combination of indexes using different index units helps the retrieval performance. Word using combination could obtain a more satisfactory result in general.

INDEXING, RETRIEVAL AND RE-RANK

Our basic idea is that documents supported by more evidences are more relevant to the query and we considered the results retrieved from different indexes as the evidences. Lemur Toolkit was adopted to create three indexes of different types. KeyFile-Unigram-Index, KeyFile-Word-Index and Indri-Word-Index. After retrieving, we re-rank the result documents by a simple interpolating algorithm.

CONCLUSION

A simple weighting scheme has been used in our IR system, which brings a slight improvement to performance. We exploit Open Category in Baidu and Related Searches in Wangfang to solve the vocabulary mismatch. Our experiments show that combination of indexes using different index units helps the retrieval performance. Word using combination could obtain a more satisfactory result in general.

Contact Information

School Of Computer Science
Inner Mongolia University
Hohhot, 010021
Inner Mongolia Autonomous Region, China
Email: cssgl@imu.edu.cn
Tel:+86-471-4992341