Merging Mechanisms in Multilingual Information Retrieval

Wen-Cheng Lin and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, TAIWAN, R.O.C.
E-mail: denislin@nlg.csie.ntu.edu.tw; hh_chen@csie.ntu.edu.tw

Abstract

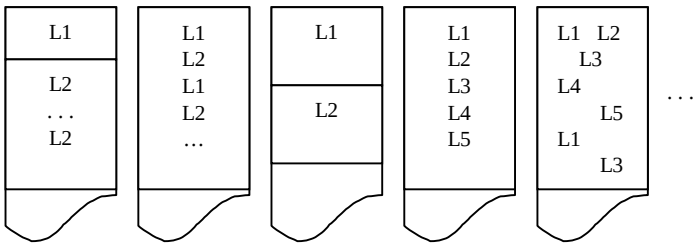
National Taiwan University (NTU) Natural Language Processing Laboratory (NLPL) participated in MLIR task in CLEF 2002. We submitted five official multilingual runs. In this paper, we try to resolve the collection fusion problem. We experimented with several merging strategies that merge the results of several intermediate runs.

1. Introduction

Multilingual Information Retrieval [4] uses a query in one language to retrieve documents in different languages. A multilingual data collection is a set of documents that are written in different languages. There are two types of multilingual data collection. The first one contains several monolingual document collections. The second one consists of multilingual-documents. A multilingual-document is written in more than two languages. Some multilingual-documents have a major language, i.e. most part of the document is written in the same language. For example, a document is written in Chinese, but the abstract is in English. Therefore this document is a multilingual-document and Chinese is its major language. The significances of different languages in a multilingual-document may be different. For example, the English translation of a Chinese proper noun is a useful clue when using English queries to retrieve Chinese documents. Therefore, the English translation should have higher weight. Figure 1 shows these two types of multilingual data collections.

In Multilingual Information Retrieval, queries and documents are in different languages. We can translate queries, or translate documents, or translate both queries and documents into an intermediate language to unify the languages of queries and documents. Figure 2 shows some MLIR architectures when query translation is adopted. The front-end controller processes queries, translates queries, submits translated queries to the monolingual IR systems, collects the relevant document lists reported by IR systems and merges them. Figure 3 shows document translation architectures.

(a) A set of monolingual document collections



(b) Some types of multilingual-documents

Fig 1. Multilingual Data Collections

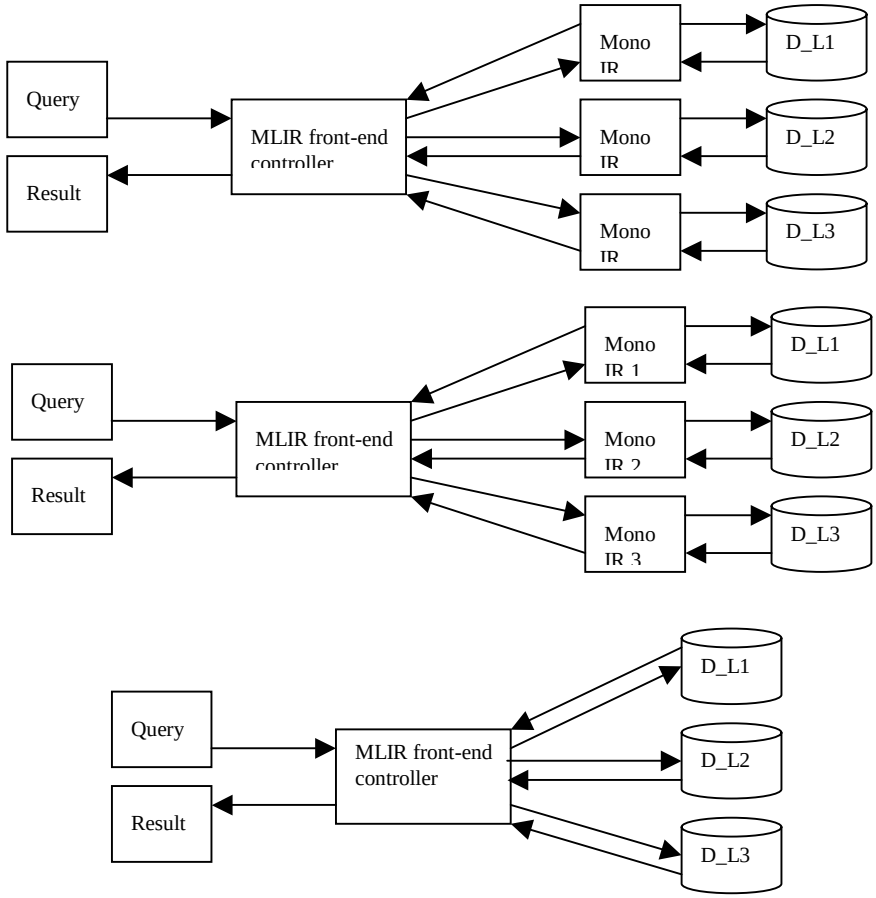


Fig 2. MLIR Architectures when Query Translation is Adopted

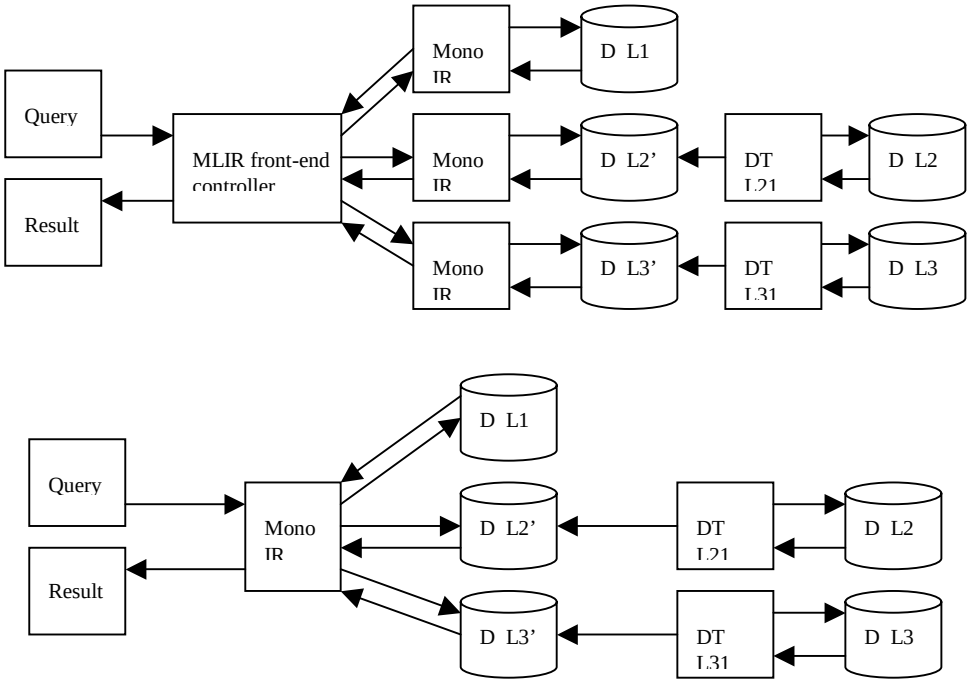


Fig 3. Document Translation Architectures

In addition to language barrier issue, how to conduct a ranked list that contains documents in different languages from several text collections is also critical. There are two architectures in MLIR: centralized and distributed. The first two architectures in Figure 2 and first architecture in Figure 3 are distributed architectures; the remaining architectures in Figures 2 and 3 are centralized architectures. In a centralized architecture, a huge collection that contains documents in different languages is used. In a distributed architecture, documents in different languages are indexed and retrieved separately. The results of each run are merged into a multilingual ranked list. Several merging strategies have been proposed. Raw-score merging selects documents based on their original similarity scores. Normalized-score merging normalizes the similarity score of each document and sorts all documents by the normalized score. For each topic, the similarity score of each document is divided by the maximum score in this topic. Round-robin merging interleaves the results in the intermediate runs. In this experiment, we adopted distributed architecture and proposed a merging strategy to merge the result lists.

The rest of this paper is organized as follows. Section 2 describes the indexing method. Section 3 shows the query translation process. Section 4 describes our merging strategies. Section 5 shows the experiment results. Section 6 concludes the remarks.

2. Indexing

The document set used in CLEF2002 MLIR task consists of English, French, German, Spanish and Italian. The numbers of documents in English, French, German, Spanish and Italian document sets are 113,005, 87,191, 225,371, 215,738 and 108,578 respectively.

The IR model we used is the basic vector space model. Documents and queries are represented as term vectors, and cosine vector similarity formula is used to measure the similarity of a query and a document. The term weighting function is tf*idf. Appropriate terms are extracted from each document in indexing stage. In the experiment, the <HEADLINE> and <TEXT> sections in English documents were used for indexing. For Spanish documents, the <TITLE> and <TEXT> sections were used. When indexing French, German and Italian documents, the <TITLE>, <TEXT>, <TI>, <LD> and <TX> sections were used. The words in these sections were stemmed, and stopwords were removed. The stopword lists and stemmers were developed by University of Neuchatel (The stopword lists and stemmers are available at http://www.unine.ch/info/clef/) [5]

3. Query translation

In the experiment, the English queries were used as source queries and translated into target languages, i.e. French, German, Spanish and Italian. In our previous experiments [1, 3], we used CO model to translate queries. CO model uses word co-occurrence information trained from a target language text collection to disambiguate the translations of query terms. In this experiment, we didn't have enough time to train the word co-occurrence information for the languages used in CLEF 2002 MLIR task. Thus, we used a simple method to translate the queries. We adopted a dictionary-based approach to translate the queries. For each English query term, we found its translation equivalents by looking up a dictionary and selected the first two translation equivalents to be the target language query terms. The dictionaries we used are Ergane English-French, English-German, English-Spanish and English-Italian dictionaries (The Ergane dictionaries are available at http://www.travlang.com/Ergane). There are 8,839, 9,046, 16,936 and 4,595 terms in Ergane English-French, English-German, English-Spanish and English-Italian dictionaries, respectively.

4. Merging strategies

There are two architectures in MLIR, i.e., centralized and distributed. In a centralized architecture, document collections in different languages are viewed as a single document collection and are indexed in one huge index file. The advantage of a centralized architecture is that it avoids the merging problem. It needs only one retrieving phase to produce a result list that contains documents in different languages. One of problems of centralized architecture is that the weights of index terms are over weighting. The total number of documents increases but the number of occurrences of a term does not. Thus, the idf of a term is increased and the weight is over-weighting. This phenomenon is acuter in small text collection. For example, the N in idf formula is 87,191 when French document is used. However, this value is increased to 749,883, i.e. about 8.6 times larger, if the three document collections are merged together. Comparatively, the weights of German index terms are increased 3.33 times due to the size of N. The increments of weights are unbalance for document collections in different size. Thus, it makes retrieval result preferring documents in small document collection.

The second architecture is a distributed MLIR. Documents in different languages are indexed and retrieved separately. The ranked lists of all monolingual and cross-lingual runs are merged into one multilingual ranked

list. How to merge result lists is a problem. Recent works have proposed various approaches to deal with merging problem. A simple merging method is raw-score merging that sorts all results by their original similarity scores and then selects the top ranked documents. Raw-score merging is based on the assumption that the similarity scores across collections are comparable. However, the collection-dependent statistics in document or query weights invalidates this assumption [2, 6]. Another approach, round-robin merging, interleaves the results based on the rank. This approach assumes that each collection has approximately the same number of relevant documents and the distribution of relevant documents is similar across the result lists. Actually, different collections do not contain equal numbers of relevant documents. Thus the performance of round-robin merging may be poor. The third approach is normalized-score merging. For each topic, the similarity score of each document is divided by the maximum score in this topic. After adjusting scores, all results are put into a pool and sorted by the normalized score. This approach maps the similarity scores of different result lists into the same range, from 0 to 1, and makes the scores more comparable. But it has a problem. If the maximum score is much higher than the second one in a result list, the normalized-score of document at rank 2 would be low even if its original score is high. Thus, the final rank of this document would be lower than that of the top ranked documents with very low but similar original scores in another result list.

Similarity score reflects the degree of similarity between a document and a query. A document with higher similarity score seems more relevant to the desired query. But, if the query is not formulated well, e.g., un-appropriate translation of a query, a document with high score still does not meet the user's information need. When merging results, such documents that have high scores should not be included in the final result list. Thus, we have to consider the effectiveness of each individual run in the merging stage. The basic idea of our merging strategy is that adjusting the similarity scores of documents in each result list to make them more comparable and to reflect their confidence. The similarity scores are adjusted by the following formula.

$$\hat{S}_{ij} = S_{ij} \times \frac{1}{\overline{S}_k} \times W_i \quad (1)$$

where S_{ij} is the original similarity score of the document at rank j in the ranked list of topic i,
$\hat{S}_{ij}$ is the adjusted similarity score of the document at rank j in the ranked list of topic i,
$\overline{S}_k$ is the average similarity score of top k documents, and
W_i is the weight of query i in a cross lingual run.

We divide the weight adjusting process into two steps. First, we use a modified score normalization method to normalize the similarity scores. The original score of each document is divided by the average score of top k documents instead of the maximum score. We call this normalized-by-top-k. Second, the normalized score multiplies a weight that reflects the retrieval effectiveness of the desired topic in each text collection. However, due to not knowing the retrieval performance in advance, we have to guess the performance of each run. For each language pair, the queries are translated into target language and then the target language documents are retrieved. A good translation should have better performance. We can predict the retrieval performance based on the translation performance. There are two factors affecting the translation performance, i.e., the degree of translation ambiguity and the number of unknown words. For each query, we compute the average number of translation equivalents of query terms and the number of unknown words in each language pair, and use them to compute the weights of each cross lingual run. The weight can be determined by the following formula:

$$W_i = c_1 + \left[c_2 \times \left(\frac{1}{T_i} \right) \right] + \left[c_3 \times \left(1 - \frac{U_i}{n_i} \right) \right] \quad (2)$$

where W_i is the weight of query i in a cross lingual run,
T_i is the average number of translation equivalents of query terms in query i,
U_i is the number of unknown words in query i,
n_i is the number of query terms in query i, and
c_1, c_2 and c_3 are tunable parameters, and $c_1+c_2+c_3=1$.

5. Results

We submitted five multilingual runs. All runs use title and description fields. The five multilingual runs use English topics as source queries. The English topics were translated into French, German, Spanish and Italian. The source English topics and translated French, German, Spanish and Italian topics were used to retrieve the corresponding document collections. Then, we merged these five result lists. We used different merging strategies for the five multilingual runs:

1. NTUmulti01

The result lists were merged by normalized-score merging strategy. The maximum similarity score was used for normalization. After normalization, all results were put in a pool and were sorted by the adjusted score. The top1000 documents were selected as the final results.

2. NTUmulti02
   In this run, we used the modified normalized-score merging method. The average similarity score of top 100 documents were used for normalization. We did not consider the performance drop down caused by query translation. That is, the weight W_i in formula (1) was 1 for every sub run.
3. NTUmulti03
   First, the similarity scores of each document were normalized. The maximum similarity score was used for normalization. Then we assigned a weight W_i to each intermediate run. The weight was determined by formula (2). The values of c_1, c_2 and c_3 were 0, 0.4 and 0.6, respectively.
4. NTUmulti04
   We used formula (1) to adjust the similarity score of each document. We used the average similarity score of top 100 documents for normalization. The weight W_i was determined by formula (2). The values of c_1, c_2 and c_3 were 0, 0.4 and 0.6, respectively.
5. NTUmulti05
   In this run, the merging strategy is similar to run NTUmulti04. The difference was that each intermediate run was assigned a constant weight. The weights assigned to English-English, English-French, English-German, English-Italian and English-Spanish intermediate runs were 1, 0.7, 0.4, 0.6 and 0.6 respectively.

The results of our official runs are shown in Table 1. The performance of normalized-score merging is bad. The average precision of run NTUmulti01 is 0.0173. When using our modified normalized-score merging strategy, the performance is better. The average precision is increased to 0.0266. Run NTUmulti03 and NTUmulti04 have considered the performance drop down caused by query translation. Table 2 shows the unofficial evaluation of intermediate monolingual and cross-lingual runs. The performance of English monolingual run is much better than cross-lingual runs. Therefore, the cross-lingual runs should have lower weights when merging results. The results show that the performances are improved by decreasing the importance of un-effective cross-lingual runs. The average precisions of runs NTUmulti03 and NTUmulti04 are 0.0336 and 0.0373, which are better than run NTUmulti01 and NTUmulti02. Run NTUmulti05 assigned constant weights to each intermediate runs. Its performance is slightly worse than run NTUmulti04. All our official runs didn't perform well. This is because that the performances of cross-lingual runs are very bad. In the experiments, we did not disambiguate the senses of query terms when translating queries and the numbers of words contained in the bilingual dictionaries we used are too few, the queries were not translated well, thus the performances of cross-lingual runs are very poor. If we use larger dictionaries and disambiguate the word senses, the performance should be better.

In order to compare the effectiveness of different merging strategies, we also conducted several unofficial runs:

1. ntu-multi-raw-score
   We used raw-score merging to merge result lists.
2. ntu-multi-round-robin
   We used round-robin merging to merge result lists.
3. ntu-multi-centralized
   This run adopted centralized architecture. All document collections were indexed in one index file. The topics contained source English query terms, and other translated query terms.

Table 1. The Results of Official Runs

Run	Average Precision	Recall
NTUmulti01	0.0173	1083 / 8068
NTUmulti02	0.0266	1135 / 8068
NTUmulti03	0.0336	1145 / 8068
NTUmulti04	0.0373	1195 / 8068
NTUmulti05	0.0361	1209 / 8068

Table 2. The Results of Intermediate Runs

Run	# Topic	Average Precision	Recall
English-English	42	0.2722	741 / 821
English-French	50	0.0497	490 / 1383
English-German	50	0.0066	201 / 1938
English-Italian	49	0.054	426 / 1072
English-Spanish	50	0.0073	223 / 2854

Table 3. The Results of Unofficial Runs

Run	Average Precision	Recall
ntu-multi-raw-score	0.0381	1180 / 8068
ntu-multi-round-robin	0.0224	1165 / 8068
ntu-multi-centralized	0.0398	1413 / 8068

The results of unofficial runs are shown in Table 3. The performance of raw-score merging is good. This is probably because we use the same IR model and term weighting scheme for all text collections. When using round-robin merging strategy, the performance is bad. The best run is ntu-multi-centralized which indexes all documents in different languages together. In this run, most top ranked documents are in English, French and Italian in most topics. The performance of English monolingual retrieval is much better than other runs. The average precisions of English-German and English-Spanish cross-lingual runs are quite low. Therefore, the result list should not contain too many German and Spanish documents. The over-weighting phenomenon in centralized architecture makes the scores of French, Italian and English documents increase. Thus, more French, Italian and English documents are included in the result list of run ntu-multi-centralized. This makes the performance better. If we use the German or Spanish queries as source queries, the performance of centralized architecture may be not so good.

6. Concluding Remarks

In the experiment, we proposed some merging strategies to integrate the result lists of collections in different languages. The results showed that the performance of our merging strategies was similar to that of raw-score merging and was better than normalized-score and round-robin merging. The performance of run NTUmulti05 was good. The weights of English-German and English-Spanish runs were not low enough. If we decrease the weights of these two runs, the performance should be better. The results showed that we could gain better performance by adjusting the similarity score appropriately. The similar results also appeared in NTCIR multilingual IR task. How to determine appropriate weights is an important issue. Considering the degree of ambiguity, i.e. lowering the weights of more ambiguous query terms, improves some performance. The centralized approach performed well. We will do more experiments to find out if the centralized approach still works when using German or Spanish queries as source queries.

References

[1] Chen, H.H., Bian, G.W., and Lin, W.C., 1999. Resolving translation ambiguity and target polysemy in cross-language information retrieval. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*, Maryland, June 1999. Association for Computational Linguistics, 215-222.

[2] Dumais, S.T., 1992. LSI meets TREC: A Status Report. In P*roceedings of the First Text REtrieval Conference (TREC-1)*, Gaithersburg, Maryland, November, 1992. NIST Publication, 137-152.

[3] Lin, W.C. and Chen, H.H., 2002. NTU at NTCIR3 MLIR Task. In *Working Notes for NTCIR3 workshop*, 2000.

[4] Oard, D.W. and Dorr, B.J., 1996. A Survey of Multilingual Text Retrieval. Technical Report UMIACS-TR-96-19, University of Maryland, Institute for Advanced Computer Studies.

[5] Savoy, J., 2001. Report on CLEF-2001 Experiments: Effective Combined Query-Translation Approach. In Evaluation of Cross-Language Information Retrieval Systems, Lecture Notes in Computer Science, Vol. 2406, Darmstadt, Germany, September, 2001. Springer, 27-43.

[6] Voorhees, E.M., Gupta, N.K., and Johnson-Laird, B., 1995. The Collection Fusion Problem. In *proceedings of the Third Text REtrieval Conference (TREC-3)*, Gaithersburg, Maryland, November, 1994. NIST Publication, 95-104.