

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
denislin@nlg.csie.ntu.edu.tw
hh_chen@csie.ntu.edu.tw

Abstract. This paper considers centralized and distributed architectures for multilingual information retrieval. Several merging strategies, including raw-score merging, round-robin merging, normalized-score merging, and normalized-by-top- k merging, were investigated. The effects of translation penalty on merging was also examined. The experimental results show that the centralized approach is better than the distributed approach. In the distributed approach, the normalized-by-top- k merging with translation penalty outperforms other merging strategies, except for raw-score merging. Because the performances of English to other languages are similar, raw-score merging gives better performance in our experiments. However, raw-score merging is not workable in practice if different IR systems are adopted.

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 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 one language. Some multilingual documents have a major language, i.e., most of the document is written in the same language. For example, a document can be 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. In this case, 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 either translate queries, or documents, or both 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 monolingual IR systems, collects the relevant document lists reported by IR systems and merges the results. Figure 3 shows another alternative, i.e., architectures for document translation.

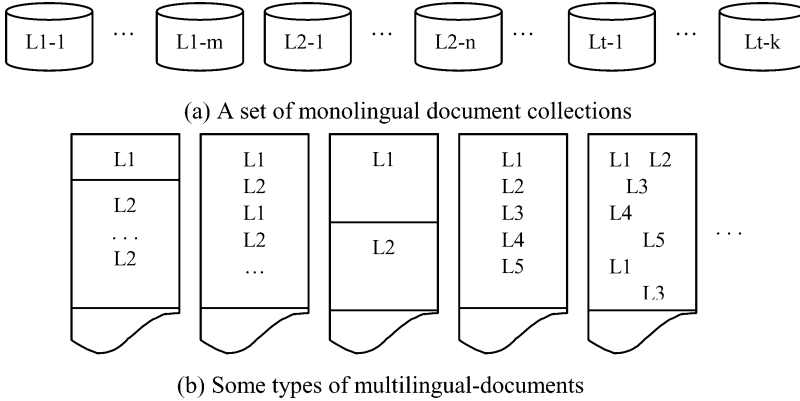


Fig. 1. Multilingual data collections

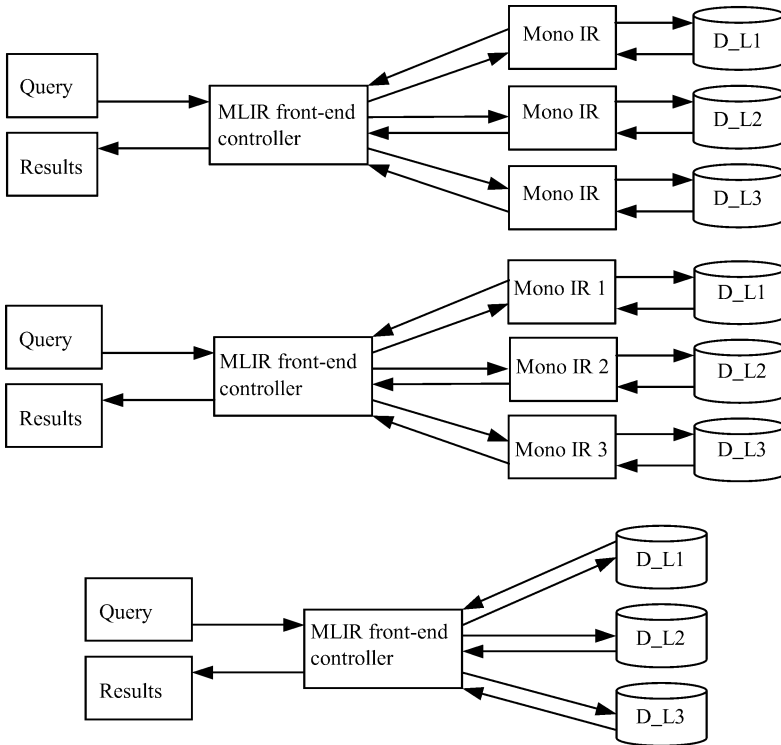


Fig. 2. Architectures of query translation

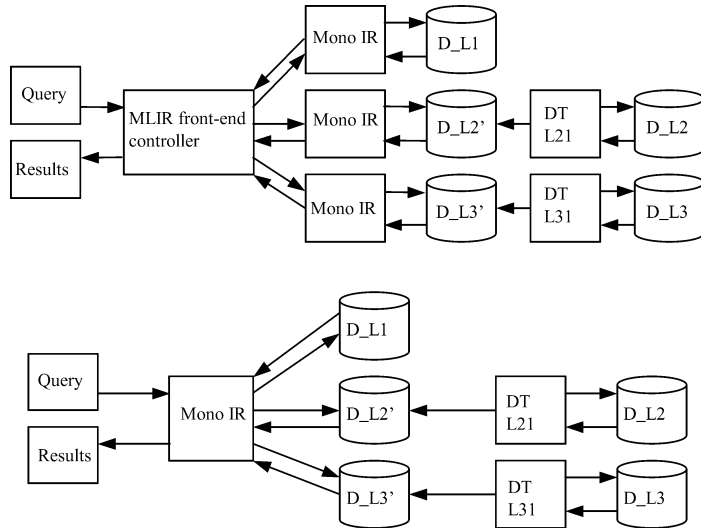


Fig. 3. Architectures for document translation

In addition to the language difference issue, the way in which a ranked list containing documents in different languages from several text collections is produced is also critical. There are two possible architectures in MLIR – let's call them centralized and distributed. The first two architectures in Figure 2 and the 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 containing 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 the 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 paper, we adopt distributed architecture and propose several merging strategies to produce 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 results of our experiments. Section 6 gives concluding 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 the 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 the 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. Stopword lists and stemmers developed by University of Neuchatel are available at <http://www.unine.ch/info/clef/> [5].

3 Query Translation

English queries were used as source queries and translated into target languages, i.e., French, German, Spanish and Italian. In the past, we used a co-occurrence (abbreviated as CO) model [1, 3] to disambiguate the use of queries. The CO model employed word co-occurrence information extracted from a target language text collection to disambiguate the translations of query terms. In the official runs, we did not have enough time to train the word co-occurrence information for the languages used in the CLEF 2002 MLIR task. Thus, we used a simple method to translate the queries. A dictionary-based approach was adopted. For each English query term, we found its translation equivalents by looking up a dictionary and considered the first two translation equivalents to be the target language query terms. The dictionaries we used are the Ergane English-French, English-German, English-Spanish and English-Italian dictionaries. They are available at <http://www.travlang.com/Ergane>. There are 8,839, 9,046, 16,936 and 4,595 terms in the Ergane English-French, English-German, English-Spanish and English-Italian dictionaries, respectively.

4 Merging Strategies

There are two possible 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 retrieval phase to produce a result list that contains documents in different languages. One of the problems of a centralized architecture is that the index terms may be over-weighted. In the traditional $tf-idf$ scheme, the idf of a query term depends on the number of documents in which it occurs. In a centralized architecture, the total number of documents increases but the number of occurrences of a term may not. In such a case, the idf of a term is increased and it is over-weighted. This phenomenon is more apparent in a small text collection. For example, the N in the idf formula is 87,191 when French document set is used. However, this value is increased to 749,883, i.e., about 8.60 times larger, if the five 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 unbalanced for document collec-

tions in different sizes. Thus, an IR system may perform better for documents in small document collections.

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 literature has proposed various approaches to deal with merging problem. A very simple merging method is the raw-score merging, which sorts all results by their original similarity scores, and then selects the top ranked documents. Raw-score merging is based on the postulation that the similarity scores across collections are comparable. However, collection-dependent statistics for document or query weights invalidates this postulation [2, 6]. Another approach, round-robin merging, interleaves the results based on the rank. This approach postulates 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 the same result list, the normalized-score of the document at rank 2 would be reduced even if its original score was 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.

The similarity score reflects the degree of similarity between a document and a query. A document with a higher similarity score seems to be more relevant to the given query. But, if the query is not formulated well, e.g., inappropriately translated, a document with a high score may still not meet a user's information need. When merging results, such documents that have incorrect high scores should not be included in the final result list. Thus, the effectiveness of each individual run should be considered in the merging stage. The basic idea of our merging strategy is that of 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 using the following formula.

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

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 ,

\bar{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 the 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 given topic in each text collection. However, as we do not know the retrieval performance in advance, we have to guess the performance of each run. For each language pair, the queries are translated into the target language and then the target language documents are retrieved. A good translation should perform better. We can predict the retrieval performance based on the translation performance. There are two factors affecting 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 formulas:

$$W_i = c_1 + \left[c_2 \times \left(\frac{51 - T_i}{50} \right)^2 \right] + \left[c_3 \times \left(1 - \frac{U_i}{n_i} \right) \right]. \quad (2)$$

$$W_i = c_1 + \left[c_2 \times \left(\frac{1}{\sqrt{T_i}} \right) \right] + \left[c_3 \times \left(1 - \frac{U_i}{n_i} \right) \right]. \quad (3)$$

$$W_i = c_1 + \left[c_2 \times \left(\frac{1}{T_i} \right) \right] + \left[c_3 \times \left(1 - \frac{U_i}{n_i} \right) \right]. \quad (4)$$

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 of Our Experiments

5.1 Official Results

We submitted five multilingual runs. All runs used the title and description fields. The five runs used 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. We then merged the five result lists. The following different merging strategies were employed.

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 top 1000 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 the top 100 documents was used for normalization. We did not consider the performance decrement 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. We then assigned a weight W_i to each intermediate run. The weight was determined by formula (4). 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, and then considered the average similarity score of the top 100 documents for normalization. The weight W_i was determined by formula (4). 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 improves. The average precision increases to 0.0266. Runs NTUmulti03 and NTUmulti04 considered the performance decrement caused by query translation. Table 2 shows the unofficial evaluation of intermediate monolingual and cross-lingual runs. The performance of the English monolingual run is much better than that of 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 those of runs NTUmulti01 and NTUmulti02. Run NTUmulti05 assigned constant weights to each of the intermediate runs. Its performance is slightly worse than that of run NTUmulti04. All our official runs did not perform well.

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	0.0540	426 / 1072
English-Spanish	50	0.0073	223 / 2854

Table 3. New results after removing a bug

Run	Average Precision	Recall
English-English (fixed)	0.2763	741 / 821
English-French (fixed)	0.1842	735 / 1383
English-German (fixed)	0.1928	972 / 1938
English-Italian (fixed)	0.1905	691 / 1072
English-Spanish (fixed)	0.1084	1021 / 2854
NTUmulti01 (fixed)	0.1206	2689 / 8068
NTUmulti02 (fixed)	0.1311	2651 / 8068
NTUmulti03 (fixed)	0.0992	2532 / 8068
NTUmulti04 (fixed)	0.0954	2489 / 8068
NTUmulti05 (fixed)	0.0962	2413 / 8068

5.2 Post-evaluation

After official evaluation, we checked every step of our experiments, and found that we made a mistake. When indexing documents, the index terms were not transformed into lower case, but the query terms were all in lower case. After fixing this bug, we obtained the new results shown in Table 3. The average precision of each run is much better than that of our official results. Normalized-by-top- k (NTUmulti02 (fixed)) is still better than normalized-score merging (NTUmulti01 (fixed)). When considering the query translation penalty, normalized-score merging is slightly better than normalized-by-top- k . We used other weighting formulas, i.e. formulas (2) and (3), for further investigations. When using formulas (2) and (3), normalized-by-top- k is better. Table 4 shows the results. Compared with NTUmulti01 (fixed) and NTUmulti02 (fixed), average precision decreased when taking performance decrement caused by query translation into consideration.

To compare the effectiveness of our approaches with previous 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 a centralized architecture. All document collections were indexed in one index file. The topics contained source English query terms, and other translated query terms.

Table 4. Normalized-score merging and normalized-by-top-100 merging with different merging weighting formulas

Merging strategy	Merging weight	Average Precision	Recall
Normalized-score merging ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1124	2683 / 8068
	formula 3	0.1049	2630 / 8068
	formula 4	0.0992	2532 / 8068
Normalized-by-top-100 ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1209	2649 / 8068
	formula 3	0.1076	2591 / 8068
	formula 4	0.0954	2489 / 8068

Table 5. Raw-score merging, round-robin merging and centralized architecture

Run	Average Precision	Recall
ntu-multi-raw-score	0.1385	2627 / 8068
ntu-multi-round-robin	0.1143	2551 / 8068
ntu-multi-centralized	0.1531	3024 / 8068

Table 6. Using new translated queries

Merging strategy	Merging weight	Average Precision	Recall	Old translation scheme
English-French		0.1857	800/ 1383	0.1842
English-German		0.2041	1023 / 1938	0.1928
English-Italian		0.1916	700 / 1072	0.1905
English-Spanish		0.1120	999 / 2854	0.1084
Raw-score merging		0.1481	2760 / 8068	0.1385
Round-robin merging		0.1169	2610 / 8068	0.1143
Normalized-score merging		0.1171	2771 / 8068	0.1206
Normalized-score merging ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1092	2763 / 8068	0.1124
	formula 3	0.1025	2688 / 8068	0.1049
	formula 4	0.0979	2611 / 8068	0.0992
Normalized-by-top-100		0.1357	2738 / 8068	0.1311
Normalized-by-top-100 ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1254	2729 / 8068	0.1209
	formula 3	0.1118	2656 / 8068	0.1076
	formula 4	0.0988	2566 / 8068	0.0954
centralized		0.1541	3022 / 8068	0.1531

Table 7. Number of English query terms without translation but in target language corpora

Run	French	German	Italian	Spanish
# query terms	427	427	427	427
# query terms without translation equivalents	153	157	215	130
# query terms without translation equivalents but in target language corpora	111	120	161	86

Table 8. Considering English query terms without translation but in the target language corpora

Merging strategy	Merging weight	Average Precision	Recall	Table 6*
Normalized-score merging				0.1171
Normalized-score merging ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1194	2792 / 8068	0.1092
	formula 3	0.1158	2765 / 8068	0.1025
	formula 4	0.1126	2747 / 8068	0.0979
Normalized-by-top-100				0.1357
Normalized-by-top-100 ($c_1=0$; $c_2=0.4$; $c_3=0.6$)	formula 2	0.1360	2778 / 8068	0.1254
	formula 3	0.1315	2751 / 8068	0.1118
	formula 4	0.1250	2704 / 8068	0.0988

The results are shown in Table 5. The performance of raw-score merging is good. This is because we use the same IR model and term weighting scheme for all text collections, and the performances of English to other languages are similar (see Table 3). When using the round-robin merging strategy, the performance is worse. The best run is ntu-multi-centralized. This run indexes all documents in different languages together.

When translating the queries, we chose the first two translation equivalents in the initial experiments. But the order of translation equivalents in a dictionary is not based on the characteristics of a corpus. We counted the occurrence frequency of each word in the test corpora and selected the first two translation equivalents with the highest frequency. The performances of cross-lingual and multilingual runs are improved by using the new translated queries. Table 6 shows the results.

From Table 6, it can be seen that the centralized architecture gives the best multilingual run. Raw-score merging is slightly worse than the centralized architecture and better than the other merging strategies. Normalized-by-top- k is better than normalized-score merging and round-robin merging. When considering the translation penalty, normalized-by-top- k is also better than normalized-score merging, but is still worse than raw-score merging. Table 6 also shows that performance decreases when considering the translation penalty. In the query translation phase, if a query term does not have any translation equivalents, the original English query term was retained in the translated query. That is, the English query terms which did not have

any translation were used to retrieve target language documents. If such an English term occurs in the target language documents, it is useful in cross-lingual information retrieval and can be considered as a word with just one translation when computing the merging weight. Table 7 lists the number of English query terms that do not have a translation, but occur in the target language corpora. As the number of query terms without a translation is less, the merging weight is higher. The results in Table 8 show that English query terms that do not have a translation but occur in the target language corpus is useful. The average precisions of runs that use formula (2) to compute merging weight (i.e., the third and the seventh rows in Table 8) are slightly better than those of runs which do not consider translation penalty (the second and sixth rows in Table 8). This shows that translation penalty is helpful if a precise translation model is adopted.

6 Concluding Remarks

This paper presents centralized and distributed architectures in MLIR. In the experiments reported, the centralized approach performed well. However, a centralized architecture is not suitable in practice, especially for very huge corpora. A distributed architecture is more flexible. It is easy to add or delete corpora in different languages and employ different retrieval systems in a distributed architecture.

The merging problem is critical in distributed architectures. In this paper, we proposed several merging strategies to integrate the result lists of collections in different languages. Normalized-by-top- k avoids the drawback of normalized-score merging. When merging intermediate runs, we also consider the performance drop caused by query translation. 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. Considering the degree of ambiguity, i.e., lowering the weights of more ambiguous query terms, improves some performance. We also employ similar experimental designs for Asian language multilingual information retrieval in NTCIR3 [3]. Similarly, we found that a centralized approach is better than a distributed approach. In a distributed approach, normalized-by-top- k with consideration of translation penalty outperforms other strategies, including raw-score merging and normalized-score merging. The trend is similar in CLEF2002 and NTCIR3 except that raw-scoring merging in CLEF2002 is better than our approach. The possible reason may be that the performances of English to documents in other languages are similar in CLEF2002, but different in NTCIR3. However, raw-scoring merging is not workable in practice if different search engines are adopted.

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 *Proceedings 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*, Tokyo, October, 2002. National Institute of Informatics.
- [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.