A Study of Learning a Merge Model for Multilingual Information Retrieval

Ming-Feng Tsai
mftsai@nlg.csie.ntu.edu.tw

Yu-Ting Wang
ytwang@nlg.csie.ntu.edu.tw

Hsin-Hsi Chen
hhchen@ntu.edu.tw

Department of Computer Science and Information Engineering
National Taiwan University
Taipei 106, Taiwan

ABSTRACT

This paper proposes a learning approach for the merging process in multilingual information retrieval (MLIR). To conduct the learning approach, we also present a large number of features that may influence the MLIR merging process; these features are mainly extracted from three levels: query, document, and translation. After the feature extraction, we then use the FRank ranking algorithm to construct a merge model; to our knowledge, this practice is the first attempt to use a learning-based ranking algorithm to construct a merge model for MLIR merging. In our experiments, three test collections for the task of crosslingual information retrieval (CLIR) in NTCIR3, 4, and 5 are employed to assess the performance of our proposed method; moreover, several merging methods are also carried out for a comparison, including traditional merging methods, the 2-step merging strategy, and the merging method based on logistic regression. The experimental results show that our method can significantly improve merging quality on two different types of datasets. In addition to the effectiveness, through the merge model generated by FRank, our method can further identify key factors that influence the merging process; this information might provide us more insight and understanding into MLIR merging.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval - Selection Process

General Terms

Design, Experimentation, Performance

Keywords

FRank, Merge Model, MLIR

1. INTRODUCTION

Multilingual information retrieval (MLIR) is usually carried out by first performing cross-language information retrieval (CLIR) on separate collections, each for a language. Once a list of monolingual results has been retrieved in each collection, all the lists are merged to produce a multilingual result list. Because of various translation and retrieval qualities in different collections, how to merge a unique result list that includes more relevant documents from different collections has become one of the major issues in MLIR.

In the literature, most traditional merging methods for MLIR are heuristic approaches such as raw-score [16], round-robin [16], normalized-by-top1 [12], and normalized-by-topk merging [8]. These heuristic methods are based on a similar assumption that relevant documents are homogeneously distributed over monolingual result lists; therefore, based on the assumption, these methods can locally adjust or normalize the scores of retrieved documents to produce a multilingual result list. However, if the assumption is invalidated, these heuristic methods have a large decrease of precision in the merging process. Instead of directly merging monolingual result lists, the 2-step merging strategy [10] uses reindexing techniques on the documents retrieved with respect to each query term to indirectly obtain a multilingual result list. Although performing well, 2-step merging is also seriously damaged by several factors such as the number of meaningful terms presenting in a collection and the proportion of relevant documents in a collection.

This paper proposes a learning approach for the MLIR merging process. To conduct the learning method, we also present a large number of features that may influence MLIR merging; these features are mainly extracted from three levels: query, document, and translation. After the feature extraction, we use a learning-based ranking algorithm, FRank [20], to construct a merge model from the extracted features. The merge model generated by FRank is then used to merge the monolingual result lists retrieved from different collections into a multilingual one. In our experiments, three CLIR test collections in NTCIR3, 4, 5 [3, 6, 7] are employed to assess the performance of our proposed method; in addition, for a comparison, several merging methods are also carried out, including the traditional merging methods, the 2-step merging strategy [10], and the merging method based on logistic regression [18]. The experimental results show that the proposed method significantly improves merging quality on two different types of datasets. In addition to the enhancement, through the generated merge model, we can also recognize the crucial features that influence MLIR merging.

SIGIR'08, July 20–24, 2008, Singapore.
Copyright 2008 ACM 978-1-60558-164-4/08/07 ...$5.00.

The main contribution of our work includes the development of a learning scheme for the MLIR merging process. To our knowledge, this study is the first attempt to use a learning-based ranking algorithm to construct a merge model for the merging process. Under this scheme, several traditional methods can be regarded as special cases of our method. For example, if a merge model is a uniform model, then the corresponding merging process acts like raw-score merging. In addition, our contribution also includes the presentation of a large number of features possibly affecting MLIR merging, and the important factor identification via the merge model generated by FRank. This information might provide us more insight and understanding into MLIR merging.

The remainder of this paper is organized as follows. In Section 2, we briefly review previous work on MLIR merging. Section 3 presents several features and describes the use of a learning-based ranking algorithm to construct a merge model. Section 4 describes evaluation metric and the details of experimental datasets. We then report and discuss experimental results. We conclude our paper and provide several directions for future work in Section 5.

2. RELATED WORK

In the literature, several heuristic methods for the MLIR merging process have been proposed. We below review these heuristic methods. Given several monolingual result lists, raw-score merging [16] directly sorts the original scores of retrieved documents to obtain a multilingual result list; round-robin merging [16] interleaves retrieved documents by their ranks only to produce a multilingual result list. Another well-known merging approaches also include the ways of using normalization techniques on the scores of retrieved documents. The main idea behind the normalized score methods is to map the scores into the same scale for a reasonable comparison. Normalized-score merging [12] uses the score of the top one document to normalize the other documents in the same list, and then sorts the normalized scores to obtain the final list. Instead of using the top one document, normalized-by-topk, uses the average score of the top k documents to produce a multilingual result list. These traditional methods are observed from our experiments to tend to have a similar performance because based on a similar assumption that relevant documents are homogeneously distributed over the monolingual result lists.

Lin and Chen [9] postulated the degree of translation ambiguity and the number of unknown words can be used to model the MLIR merging process. They presented the following formulas to predict the effectiveness of MLIR:

$$W_i = c_1 * \left(\frac{1}{\sqrt{T_i}}\right) + c_2 * \left(1 - \frac{U_i}{n_i}\right);$$

$$W_i = c_1 * \left(\frac{1}{T_i}\right) + c_2 * \left(1 - \frac{U_i}{n_i}\right),$$

where W_i is the merging weight of query i in a CLIR run, T_i is the average number of translation 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 and c_2 are tunable parameters such that $c_1 + c_2 = 1$. However, by means of heuristic methods, it is quite difficult to obtain the optimal parameters for the above formulas.

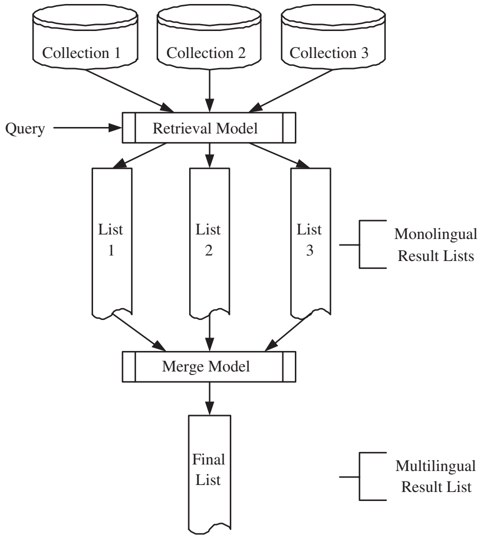

Figure 1: Traditional Framework for MLIR.

Martínez-Santiago et al. [10] proposed a 2-step merging strategy to produce a multilingual result list. Instead of directly merging the monolingual result lists into a multilingual one, 2-step merging uses re-indexing techniques on the documents retrieved with respect to each query term, and then employs the re-indexed dataset to produce a multilingual result list. By the re-indexing techniques, 2-step merging can globally consider relevant terms within data collections. Although performing well, the 2-step merging strategy would seriously be damaged by some factors such as the number of meaningful terms presenting in a collection and the proportion of relevant documents in a collection.

The related work also includes the studies on collections fusion and search results merging in distributed information retrieval (DIR). Although DIR environments tend to be monolingual and uncooperative, some related techniques have been applied for MLIR merging with several degree of success. Si and Callan [17, 19] proposed a semisupervised learning solution for the DIR merging problem. Savoy [11, 15] proposed a merging approach based on logistic regression for predicting the probability of binary relevance according to a set of independent variables. Although also based on learning techniques, these methods lack the consideration of dataset with multiple relevance judgments. Therefore, we propose a novel merging method for MLIR merging by using a learning-based ranking algorithm, in which multiple relevance judgments can be considered.

3. LEARNING A MERGE MODEL

Figure 1 illustrates a traditional framework for MLIR, in which each collection is a monolingual collection. This figure shows that the traditional MLIR is typically carried out by first performing CLIR on separate collections. Once a monolingual result list has been retrieved in each collection, all the lists are merged into a multilingual result list. Thus, the traditional MLIR framework consists of two models: one is retrieval model for retrieving documents from each monolingual collection; another is merge model for merging all the monolingual result lists into a multilingual one.

In the traditional MLIR framework, the retrieval model is usually set as several standard IR techniques such as $bm25$ [13] and cosine similarity. Moreover, in recent years, several methods [1, 2, 4, 5, 20] based on learning techniques have also been proposed for the retrieval model. As for the merge model, conventional merging methods are mostly based on some heuristics. Therefore, this paper attempts to propose a learning scheme to generate a merge model. Under this learning scheme, several conventional merging methods can be regarded as special cases of our method. For example, if the generated merge model is a uniform model, then our method acts like raw-score merging. Furthermore, our method is similar to round-robin merging if the generated merge model predicts the merging weight of a document by means of its reciprocal rank.

Below, we describe how to use a ranking algorithm based on leaning techniques to construct a merge model. In addition, we also present several features possibly affecting the MLIR merging process. Through the merge model generated by the learning-based ranking algorithm, we expect to identify critical features that really influence MLIR merging.

3.1 Feature Set

Table 1 lists the features used to construct a merge model in this study. There are 62 features extracted from three levels: query, document, and translation; moreover, all the features are represented by real numbers in our experiments. According to the extraction level, we describe these features in detail as follows.

On document level, only two features are extracted in this study. These two features are document length and title length; they represent the number of words in a document and in a document title, respectively. We consider these two features mainly because of their abilities of indicating the amount of information within a document. In this study, no retrieval feature is used to construct a merge model, although retrieval features (e.g., tf and idf) can also be regarded as document-level features. Our experiments show that the retrieval features, if included, usually tend to dominate the generated merge model; as a result, this situation would lead to a difficulty in identifying the important features affecting MLIR merging. To emphasize the merging process, therefore, we use only document length (DLength) and title length (TLength) to construct a merge model.

For query-level feature extraction, we first manually classify the terms within a query into several pre-defined categories, and then extract query-level features according to these categories. In our experiments, we primarily consider the query terms as proper names since they usually play an important role in retrieving documents. For a query, therefore, each query term is labeled as one of the following categories:

- Location/country names (Loc): e.g., Japan, Barents Sea, and Pyongyang;
- Organization names (Org): e.g., Time Warner, Amnesty International, and Los Alamos National Laboratory;
- Personal names/titles/alias (PN): e.g., Wen Ho Le, Birdman, and The Comfort Women;
- Event names (EN): e.g., Inter-Korea Summit, French Open Pairs, and Guinness World Records;
- Technical terms (TT): e.g., Anthrax, Bacillus Anthracis, and Operating System.

In addition, we also use an additional category: named entity (NET), which contains all the query terms classified into the above categories. For those terms unable to be precisely labeled, we simply classify them into two categories: concrete nouns (CN) and abstract nouns (AN). Two types of verbs, i.e., intransitive (IV) and transitive (TV), are also considered in this study. Two query examples selected from our experimental datasets will be shown in Section 4.4; these two examples consist of their query descriptions and query terms with the corresponding labels.

After the above labeling, we then extract query-level features from the labeled dataset. For a query, the feature set comprises the number of query terms (#QT) and compound words (#CW). In addition, the feature set also consists of the number of the query terms classified into the pre-defined categories (e.g., #PPN and #Loc), and the corresponding percentage with respect to total query terms (e.g., %PPN and %Loc). These query-level features are extracted mainly because we consider different types of query terms would influence the MLIR merging process differently. Through these features, therefore, we expect to realize the relation between query difficulty and merging performance.

On translation level, we extract several features capable of indicating the translation quality of a query for a language. The translation-level features include the languages used in a query and in a document (i.e., QLanguage and DLanguage); the values of these two features are set to 0 for English, 1 for Chinese, and 2 for Japanese in our experiments. In addition, the translation-level features also consist of the size of a bilingual dictionary used for various language (i.e., DictSize) and the average number of translation equivalents within a query (i.e., AvgTAD). For instance, for a language, if a query has two query terms both with three translation equivalents, then the value of AvgTAD of the query is $(3 + 3)/2 = 3$.

Furthermore, the translation-level features also consist of the number of translatable query terms (i.e., #TQT), the number of translatable compound words (i.e., #TCW), and the corresponding ratios to total query terms (i.e., %TQT and %TCW). A query with more highly ambiguous query terms, whose number of translation equivalents ≥ 3, usually tends to be not well-translated. Therefore, the translation-level features also include the number of highly ambiguous terms (i.e., #HAT) and the corresponding ratio to total query terms (i.e., %HAT). In addition to translatable query terms, we also consider the features of out of vocabulary terms, such as #OOV and %OOV. The idea of OOV features is also extended to the terms classified into various categories, thereby generating the features such as #OLoc and %OLoc. Using these translation-level features, we expect to realize the effect of translation quality to merging performance. In the next subsection, we describe a learning-based ranking algorithm to construct a merge model. Through the merge model generated by the ranking algorithm, we can realize the effects of these extracted features to MLIR merging.

Table 1: The Set of Features for the Construction of a Merge Model

Feature	Description	Feature	Description
Document Level			
TLength	Title length	DLength	Document length
Translation Level			
QLanguage	Query language	DLanguage	Document language
AvgTAD	Average translation ambiguity degree	DictSize	Size of bilingual dictionary
#TQT	# of translatable query terms	%TQT	#TQT/#QT
#HAT	# of highly ambiguous terms*	%HAT	#HAT/#TQT
#TCW	# of translatable compound words	%TCW	#TCW/#QT
#OOV	# of OOV query terms	%OOV	#OOV/#QT
#OTCW	# of OOV compound words	%OTCW	#NTCW/#QT
#OPPN	# of OOV proper names	%OPPN	#OPPN/#PPN
#OLoc	# of OOV location names	%OLoc	#OLoc/#Loc
#OOrg	# of OOV organization names	%OOrg	#OOrg/#Org
#OPN	# of OOV personal names	%OPN	#OPN/#PN
#OEN	# of OOV event names	%OEN	#OEN/#EN
#OTT	# of OOV technical terms	%OTT	#OTT/#TT
#ONET	# of OOV named entities	%ONET	#ONET/#NET
#OAN	# of OOV abstract nouns	%OAN	#OAN/#AN
#OCN	# of OOV concrete nouns	%OCN	#OCN/#CN
#OTV	# of OOV transitive verbs	%OTV	#OTV/#TV
#OIV	# of OOV intransitive verbs	%OIV	#OIV/#IV
Query Level			
#QT	# of query terms	#CW	# of compound words
#PPN	# of proper names	%PPN	#PPN/#QT
#Loc	# of location names	%Loc	#Loc/#QT
#Org	# of organization names	%Org	#Org/#QT
#PN	# of personal names	%PN	#PN/#QT
#EN	# of event names	%EN	#EN/#QT
#TT	# of technical terms	%TT	#TT/#QT
#NET	# of named entities	%NET	#NET/#QT
#AN	# of abstract nouns	%AN	#AN/#QT
#CN	# of concrete nouns	%CN	#CN/#QT
#TV	# of transitive verbs	%TV	#TV/#QT
#IV	# of intransitive verbs	%IV	#IV/#QT

* Highly ambiguous term: a term with translation equivalents ≥ 3.

3.2 The Construction of a Merge Model

The FRank ranking algorithm [20] is adopted to construct a merge model in this paper. FRank is a learning-based ranking algorithm with a novel loss called fidelity loss based on RankNet's probabilistic ranking framework [1]. Because of the helpful properties in the fidelity loss, FRank is well-suited to ranking applications with multiple relevance judgments. In addition, since our experimental collection has four relevance judgments, merging on such a collection can be regarded as a ranking application. For the details of FRank, please refer to [20].

According to FRank's generalized additive model, a merge model can be represented as:

$$M_t(x) = \sum_t \alpha_t m_t(x),$$

where $m_t(x)$ is a weak learner, α_t is the learned weight of $m_t(x)$, and t is the number of selected weak learners. Thus, when obtaining the merge model generated by FRank, we can examine the effect of a feature to MLIR merging by means of its learned weight α_t. Upon completion of the merge model, we combine it with a retrieval model by using linear combination. In our experiments, the retrieval model is set as $bm25$; then the proposed method can be represented as:

$$(1 - \lambda) * M_t(x) + \lambda * bm25,$$

where λ is the combination coefficient of $bm25$. In the above model, the number of selected weak learners t and the combination coefficient λ are two tunable parameters that can be determined on a validation dataset.

In our method, the merge model $M_t(x)$ generated by FRank can be regarded as a supplementary model to enhance the merging quality of $bm25$. This practice is consistent with the traditional procedure of MLIR, in which a retrieval model is for retrieving documents on each monolingual collection, and a merge model is for merging all the monolingual result lists into a multilingual one. Moreover, this practice also provides us a way independent from the retrieval model to examine the critical features influencing MLIR merging.

Table 2: The Details of Experimental Collections

	Number	Number of Documents		
	of	in Each Language		
Collection	Queries	C	J	E
NTCIR3	50	381,681	220,078	22,927
NTCIR4	60	381,681	596,058	347,550
NTCIR5	50	901,446	858,400	259,050

4. EXPERIMENTS

In this section, we first describe the details of experimental collections and settings, including the evaluation metric and the comparison methods used in our experiments. Moreover, we also state the procedures of building datasets for training and testing. Then, we describe the experiments of using FRank to construct a merge model and report the results of all the comparison methods. Finally, to further identify the crucial features that influence MLIR merging, we provide several discussions as to the features found in FRank.

4.1 Experimental Datasets and Settings

In our experiments, three CLIR test collections in NTCIR3, 4, and 5 [3, 6, 7] were used to build datasets for MLIR merging. These collections are mainly for the evaluation of CLIR performance on four languages: Chinese (C), Japanese (J), Korean (K), and English (E). With respect to a query, documents within the collections are labeled with four relevance judgments, including highly relevant, relevant, partially relevant, and irrelevant. Because of lack of Korean resources, we only used CJE documents to build experimental datasets. Table 2 lists the details of the CLIR test collections.

We here describe how to build datasets for the MLIR merging experiments. For each collection, we used English topics as source queries to retrieve English, Chinese, and Japanese documents; that is, there were three retrieval processes: E-E monolingual retrieval, and E-C and E-J crosslingual ones. In addition, query terms were mainly composed of the terms in the concept field of a topic description. When translating an English query term for crosslingual retrieval, we chose two translation candidates with the highest frequency in the corresponding language corpus. For example, if an English query term has three Chinese translation candidates, we select the top two candidates that appear frequently in Chinese corpus. After the E-E, E-C, and E-J retrieval processes, we then obtained three lists of monolingual results for a query; for each monolingual result list, we only used the documents whose similarity scores are not zero to construct the experimental datasets.

After obtaining the experimental datasets, we then conducted experiments on the datasets to merge the three monolingual result lists into a multilingual one. For a comparison, several merging methods were carried out in our experiments, including raw-score, round-robin, normalized-by-top1, normalized-by-topk, and 2-step merging. Among these methods, we refer to the raw-score, round-robin, normalized-by-top1, and normalized-by-topk merging methods as traditional ones. Instead of directly merging the monolingual result lists, the 2-step merging strategy uses re-indexing techniques to indirectly obtain the multilingual result list; therefore, we regard this strategy as a variant merging approach.

Table 3: The Percentage of Retrieved Relevant Documents to Total Retrieved Relevant Documents in Different Experimental Datasets

	Language		
Dataset	C	J	E
Training (NTCIR5)	21.53%	39.89%	38.57%
Testing (NTCIR3)	33.45%	27.20%	39.34%
Testing (NTCIR4)	12.68%	46.34%	40.97%

In addition to the heuristic methods, two learning-based merging methods are also performed, including one based on logistic regression [18] and ours based on the FRank ranking algorithm. The corresponding settings for learning-based methods are described in Section 4.2.

Traditional IR measure, Mean Average Precision (MAP), was used as evaluation metric in our experiments. Given a result list for query q_i, the average precision for q_i can be defined as follows:

$$\mathrm{AP}_i = \frac{\sum_{j=1}^{N}(P(j) \times pos(j))}{\#\text{ of relevant docs in } q_i},$$

where N is the number of documents retrieved, $P(j)$ is the precision value at position j, and $pos(j)$ is a binary function indicating whether the document at position j is relevant. Once all APs for all queries have been obtained, the MAP can be calculated by averaging the APs over all queries. In our experiments, the position j was set to 1000, and the $pos(j)$ is set to true if the document at position j is labeled as highly relevant or relevant; this criterion setting is consistent with the rigid measure of NTCIR's CLIR task.

4.2 Experiments of Learning a Merging Model

Table 3 lists the percentage of retrieved relevant documents to total retrieved relevant documents in different experimental datasets. Regarding the distribution of retrieved relevant documents, NTCIR3 and NTCIR5 is more balanced than NTCIR4; in addition, as indicated in Table 2, the number of articles in NTCIR5 is also more than those in NTCIR3 and NTCIR4. Due to the above reasons, the monolingual result lists in NTCIR5 were chosen as training dataset, and those in NTCIR3 and NTCIR4 as two separate testing datasets; this practice assists us to further examine the performance of the comparison methods on two different types of datasets. In addition, for each testing datasets, we also selected 10 queries as a validation to determine the parameters within the learning-based merging methods.

For the method based on logistic regression, we directly used the binary code of mySVM [14]. The parameter c within mySVM was tuned on the validation datasets; according to the validation results, the parameters c were set to 2.5 for the NTCIR3 testing dataset and 2.0 for NTCIR4. For our method, we implemented the FRank ranking algorithm [20] to construct a merge model from the proposed 62 features; in addition, all the features were normalized to the values between 0 and 1 in the experiments. The parameters of our methods were also tuned on the validation datasets; according to the results, the number of weak learners t and the combination coefficient λ were set to 7 and 0.07 for NTCIR3 as well as 13 and 0.01 for NTCIR4.

Table 4: Experimental Results on Testing Datasets

	MAP	
Merging Strategy	NTCIR3	NTCIR4
Raw-score (baseline)	0.174	0.278
Round-robin	0.174	0.180
Normalized-by-top1	0.172	0.154
Normalized-by-topk	0.175	0.173
2-step	**0.210 (0.048)**	0.246
Logistic regression	0.189 (0.348)	0.152
Our method	**0.222 (7.7e-3)**	**0.364 (3.7e-5)**

Table 4 lists the experimental results of all comparison methods on two testing datasets. Numbers in brackets indicate the *p*-value from a paired *t*-test. Bold faced numbers indicate that the entry is statistically significant from the run of raw-score merging (baseline) on the same dataset at 95% confidence level. On the balanced NTCIR3 testing dataset, the performance of the traditional merging methods is similar to that of the baseline; in comparison, our method and the 2-step merging strategy both significantly outperform the baseline. The improvement of merging method based on logistic regression, however, fails to pass the significance test. On the unbalanced NTCIR4 testing dataset, in contrast to other merging methods, ours also significantly outperforms the baseline; this consequence arises main because our method can overcome the problem of unbalanced distribution by means of FRank's ability of coordinating the proposed features.

For our method, we also conducted an experiment to further examine the effect of combination coefficient λ to merging performance. Figure 2 illustrates that the results of two baselines and those of our method with different λ. Regardless of the performance on NTCIR3 or NTCIR4, our method with various λ is constantly above raw-score merging, except for $\lambda = 0$; this exception is due to the fact that, when $\lambda = 0$, our method uses only merge model to produce a multilingual result list. Furthermore, as observed from the figure, the merging process of our method is the same as raw-score merging when $\lambda = 1$. From this point of view, the proposed merge model can be considered a supplementary model to enhance the merging quality of raw-score merging.

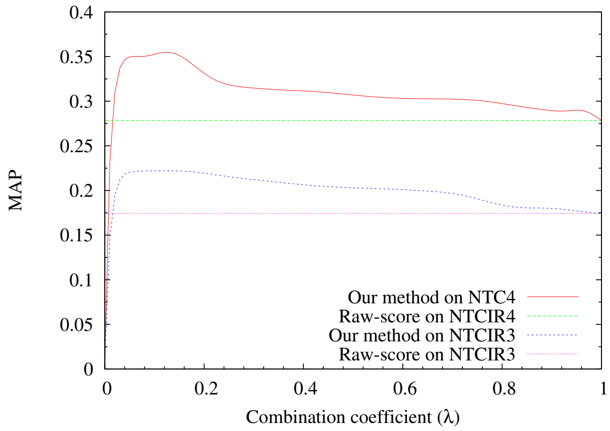

Figure 2: The Experimental Results of Our Method using Different Combination Coefficient λ

4.3 Discussions

According to different types of merging methods, we offer some observations and discussions about the experimental results as follows.

- Operating on the balanced NTCIR3 dataset, the traditional merging methods have a similar performance; on the other hand, on the unbalanced NTCIR4 dataset, these methods perform differently. This consequence arises mainly because the traditional methods are all based on a similar assumption that relevant documents are homogeneously distributed across different language collections. Therefore, when working on a dataset with balanced distribution, the traditional methods tend to obtain a similar performance. In the NTCIR4 testing dataset, however, the number of retrieved relevant documents in English and Japanese is excessively more than that in Chinese, thereby causing the situation of unbalanced distribution. Furthermore, the methods such as round-robin or normalized score, as described in [10], especially present less awareness of this situation because the score of each document is discarded (round-robin merging by rank only) or is calculated and normalized locally. Thus, on the unbalanced NTCIR4 testing dataset, the performance of round-robin and normalized score is relatively lower than that of raw-score merging, as indicated in Table 4. For the query (Qid 49) in Table 5, the average precision is 0.548 for raw-score, 0.342 for round-robin, 0.315 for normalized-by-top1, and 0.313 for normalized-by-topk.

- By means of re-indexing techniques, the 2-step merging strategy can globally consider relevant terms in different language collections; therefore, this strategy significantly outperforms the baseline on the NTCIR3 testing dataset. However, on the NTCIR4 dataset, this strategy goes under the baseline, since also damaged by the unbalanced situation. In contrast to the traditional methods, 2-step merging appears to have more sensibility to the unbalanced situation, as indicated in Table 4. For the query (Qid 49) in Table 5, the average precision of 2-step merging is 0.564, which is relatively higher than that of the traditional methods.

- For the learning-based merging methods, the method based on logistic regression has an ineffective performance in the merging process on two testing datasets. This ineffectiveness occurs mainly because logistic regression lacks the consideration of the testing datasets with four relevance judgments; thus, this indicates that logistic regression appears to be an inappropriate model for ranking applications like this work. As for our method, with FRank's ranking ability, the proposed merge model effectively coordinates the features that influence the merging process. Therefore, as indicated in Table 4, our method significantly outperforms the baseline on two testing datasets, especially on the unbalanced NTCIR4 dataset. For the query (Qid 49) in Table 5, the average precision is 0.477 for logistic regression and 0.68 for our proposed method.

Table 5: Two Query Examples in the NTCIR4 Testing Dataset

	Distribution				
Qid	E	C	J	Query description	Query terms
15	351	39	19	Find expert critical opinion on the International Monetary Fund's (IMF) policy on Asian countries.	International Monetary Fund (Org), foreign exchange crisis (EN), economic crisis (EN), Asia (Loc), IMF (Org), influence (TV)
49	207	30	195	Find articles on the actions of President Habibie concerning the East Timor Issue.	Habibie Administration (AN), Pro-Indonesia Forces (AN), President Habibie (PN), independence issue (AN), local referendum (EN), East Timor (Loc), Indonesia (Loc)

Table 6: The Most Effective Features

Feature	Total Weight	Average Weight	Level
%TQT	2.268	0.756	TL
%AN	2.526	0.631	QL
#TCW	1.076	0.538	TL
%TT	1.305	0.261	QL
%Loc	0.956	0.191	QL
TLength	0.769	0.128	DL
%PPN	0.332	0.110	QL
DLength	-0.214	-0.023	DL
%EN	-1.476	-0.184	QL
#HAT	-1.215	-0.202	TL
DLanguage	-0.250	-0.250	TL
%Org	-1.791	-0.255	QL
#OOrg	-0.847	-0.847	TL
#OTV	-1.325	-1.325	TL
#ONET	-2.209	-2.209	TL

4.4 Feature Analysis

Table 5 lists two query examples in the NTCIR4 testing dataset. Each query example in the table consists of query id, query description, labeled query terms, and the distribution of retrieved relevant documents. In addition, Table 6 lists the most effective features found in the first 100 iterations of FRank; according to their average learned weight, these features are selected, including the top 7 positive ones and 8 negative ones. Through these selected features, we attempt to find more clues about the MLIR merging process. On the basis of the extraction level, we below provide the analysis of the selected features.

- Document-level features:
  Two document-level features, i.e., TLength (+)[1] and DLength (−), are both selected as the most effective features in our experiments. Their weights indicate that with larger title length, a document has more merging weights; however, with larger document length, a document has fewer merging weights. This situation is similar to that of retrieving documents, in which document length is typically used to normalize the similarity score of a document. Therefore, we can consider that a document with larger document length usually tends to have more noise information for the processes of retrieving and merging documents.

- Query-level features:
  As indicated in Table 6, there are 6 query-level features selected as the most effective features in our experiments, including %AN (+), %TT (+), %Loc (+), %PPN (+), %EN (−), and %Org (−). This consequence occurs mainly because in our experimental datasets, a query with more abstract nouns, technical and location terms usually tends to be well-translated; however, a query with more event and organization names tends to be poorly-translated. For the merging process, therefore, a well-translated query is supposed to have more merging weights; in contrast, a poorly-translated query has fewer merging weights. Take an example in Table 5 that the query terms in Qid 49, such as Habibie Administration (AN) and Indonesia (Loc), are relatively easier to be translated by our dictionary than those terms in Qid 15, such as International Monetary Fund (Org) and foreign exchange crisis (EN); therefore, our merge model will assign more merging weights to the simple[2] Qid 49 than the difficult Qid 15. From this point of view, we consider our merge model a suitable model to discover the relation between query difficulty and merging performance.

- Translation-level features:
  As indicated in Table 6, there are 7 translation-level features selected in our experiments, including %TQT (+), #TCW (+), #HAT (−), #OOrg (−), #OTV (−), #ONET (−), and DLanguage (−). These features indicate that with more translatable query terms or compound words, a translated query for a language has more merging weights; however, with more highly ambiguous terms or OOV terms as named entities, transitive verbs, and organization names, a translated query has fewer merging weights. In addition, as indicated in the table, the weights of these translation-level features are relatively higher than those of other level features; therefore, we think the key factors influencing MLIR merging are translatable query terms, compound words, and the OOV terms as named entities, transitive verbs, and organization names. This information gives us a principled way of dealing with translation procedure for conducting a crosslingual retrieval; for example, we will concentrate on the translation quality of query terms as named entities when conducting a crosslingual retrieval. The negative sign of document language (DLanguage) is due to its value settings, thereby indicating our merge model prefers English documents to Chinese and Japanese ones.

[1] Symbols in brackets indicate the sign of the average learned weight of a feature

[2] The average precision is 0.68 for Qid 49 and 0.5 for Qid 15.

5. CONCLUSIONS

The contribution of this work includes the proposition of a learning approach for the MLIR merging process. We use the FRank ranking algorithm to construct a merge model for merging monolingual result lists into a multilingual one. Experimental results demonstrate that, even performing on a dataset with the unbalanced distribution of relevant documents, the proposed merge model can significantly improve merging quality. Moreover, the contribution of this work also includes the presentation of several features affecting MLIR merging, and the feature analysis via the merge model generated by FRank. In conclusion, the merge model indicates that for MLIR merging, the key factors are the number of translatable terms and compound words; in addition, the number of OOV terms as named entities, transitive verbs, and organization names also plays an important role in the merging process. This information provides us more insight and understanding into the MLIR merging process. Several research directions remain for future work:

- For the construction of a merge model, we would like to use other learning-based ranking algorithms such as RankSVM [5] and RankNet [1].
- For the identification of critical features, we would like to extract more representative features to construct a merge model, such as linguistic features.
- In addition, through the merge model generated by the learning-based ranking algorithm, we also expect to discover more relations within query terms, such as query term association and substitution.

6. ACKNOWLEDGMENTS

Research of this paper was partially supported by Excellent Research Projects of National Taiwan University (96R0062-AE00-02).

7. REFERENCES

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

[2] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[3] K. H. Chen, H. H. Chen, N. Kando, K. Kuriyama, S. Lee, S. H. Myaeng, K. Kishida, K. Eguchi, and H. Kim. Overview of CLIR Task at the Third NTCIR Workshop. *NTCIR-3 Proceedings*, 2003.

[4] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An Efficient Boosting Algorithm for Combining Preferences. *Journal of Machine Learning Research*, 4(6):933–969, 2004.

[5] T. Joachims. Optimizing search engines using clickthrough data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[6] K. Kishida, K. H. Chen, S. Lee, K. Kuriyama, N. Kando, H. H. Chen, and S. H. Myaeng. Overview of CLIR Task at the Fifth NTCIR Workshop. *Proc. of the NTCIR-5 Workshop Meeting*, pages 1–38, 2005.

[7] K. Kishida, K. H. Chen, S. Lee, K. Kuriyama, N. Kando, H. H. Chen, S. H. Myaeng, and K. Eguchi. Overview of CLIR task at the fourth NTCIR workshop. *Proceedings of NTCIR*, 4, 2004.

[8] W. C. LIN and H. H. CHEN. Merging mechanisms in multilingual information retrieval. *Lecture notes in computer science*, pages 175–186, 2003.

[9] W. C. LIN and H. H. CHEN. Merging results by predicted retrieval effectiveness. *Lecture notes in computer science*, pages 202–209, 2004.

[10] F. Martínez-Santiago, L. Ureña-López, and M. Martín-Valdivia. A merging strategy proposal: The 2-step retrieval status value method. *Information Retrieval*, 9(1):71–93, 2006.

[11] C. Peters. *Cross-Language Information Retrieval and Evaluation: Workshop of the Cross-Language Evaluation Forum, CLEF 2000, Lisbon, Portugal, September 21-22, 2000: Revised Papers*. Springer, 2001.

[12] A. Powell, J. French, J. Callan, M. Connell, and C. Viles. The impact of database selection on distributed searching. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–239, 2000.

[13] S. Robertson and S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, 1994.

[14] S. Ruping. mySVM-Manual. *University of Dortmund, October*, 2000.

[15] J. Savoy. Cross-language information retrieval: experiments based on CLEF 2000 corpora. *Information Processing and Management*, 39(1):75–115, 2003.

[16] J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the TREC-5 experiment: Data fusion and collection fusion. *The Fifth Text REtrieval Conference (TREC-5)*, pages 489–502, 1997.

[17] L. Si and J. Callan. A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems (TOIS)*, 21(4):457–491, 2003.

[18] L. Si and J. Callan. CLEF 2005: Multilingual Retrieval by Combining Multiple Multilingual Ranked Lists. *Sixth Workshop of the Cross-Language Evaluation Forum, CLEF*, 2005.

[19] L. Si and J. Callan. Modeling search engine effectiveness for federated search. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 83–90, 2005.

[20] M. F. Tsai, T. Y. Liu, T. Qin, H. H. Chen, and W. Y. Ma. FRank: a ranking method with fidelity loss. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 383–390, 2007.