CLEF 2005: Multilingual Retrieval by Combining Multiple Multilingual Ranked Lists

Luo Si and Jamie Callan
Language Technology Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
{lsi,callan}@cs.cmu.edu

Abstract: We participated in two tasks: Multi-8 two-years-on retrieval and Multi-8 results merging. For our multi-8 two-years-on retrieval work, simple multilingual ranked lists are first built by merging ranked lists of different languages that are generated by single types of retrieval algorithms. Then, algorithms are proposed to combine these simple multilingual ranked lists into a single ranked list. Empirical study shows that the approach of combining multilingual retrieval results can substantially improve the accuracies over single multilingual ranked lists.

Multi-8 results merging task is our primary interest. This task is viewed as similar to the results merging task of federated search. Query-specific and language-specific models are proposed to calculate comparable document scores for a small amount of documents and estimate logistic models by using information of these documents. The logistic models are used to estimate comparable scores for all documents and thus the documents can be sorted into a final ranked list. A set of experiments demonstrated the advantage of the query-specific and language-specific models against several other alternatives.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: H.3.3 Information Search and Retrieval.

General Terms

Algorithms, Experimentation.

Keywords

Multilingual retrieval, Crosslingual retrieval, Results merging.

1. Introduction

The first task as Multi-8 two-years-on is a multilingual retrieval [5,6,9,13,14,15] task, which is to search documents in eight languages with queries in a single language (i.e., English queries in this work). One method is to tune accurate bilingual retrieval results [14,15] (or monolingual results of documents in the same language as queries) and then merge the bilingual retrieval results together. For each bilingual run, previous research [5,14,15] has demonstrated how to do many instances of bilingual retrieval by tuning the methods of translating the query into target language and then generate an accurate bilingual run. Finally, those bilingual runs generated with different methods are merged into a final multilingual ranked list. However, it may not be easy to merge accurate bilingual retrieval results into accurate multilingual retrieval results. One reason is that the ranges and distributions of document scores within these bilingual ranked lists can be very different as quite different retrieval methods have been tuned to generate accurate bilingual results of different languages separately [14,15]. Therefore, it is difficult to merge those bilingual result lists with quite different characteristics.

One alternative approach of generating multilingual retrieval result is to first generate simple bilingual runs by same type of retrieval algorithm with the same configuration and then merge the bilingual results into a simple multilingual ranked list. Many simple multilingual results can be obtained by applying different retrieval algorithms with different retrieval configurations. Finally, those simple multilingual ranked lists can be combined into a more accurate multilingual ranked list. This method has been shown to be effective in the work of [5] by combing multilingual results from retrieval methods based on query translation and document translation. The multilingual retrieval system described in this work focuses on generating multilingual retrieval results by simple retrieval algorithms and also on combining several multilingual retrieval lists together into a final ranked list of high accuracy. In this work, we have proposed several methods to combine multilingual retrieval results. The empirical study shows that the approach of combining multilingual retrieval results can substantially improve the accuracies over single multilingual ranked lists.

The second task as Multi-8 results merging task is to merge ranked lists of eight different languages (i.e., bilingual or monolingual) into a single final list. This task is very similar to the results merging task of federated search [16], which merges multiple ranked lists from different web resources into a single list. Results merging task is our primary interest and our goal is to investigate the effectiveness of applying similar results merging algorithms as federated search task and compare their accuracies with other results merging algorithms.

Previous research in [14,15] has proposed to build logistic models to estimate probabilities of relevance for all documents in bilingual ranked lists by their ranks and document scores in these bilingual lists. This method is studied in this paper and a new variant of this method is proposed to improve the merging accuracy. These methods are language-specific methods as they build different models for different languages to estimate the probabilities of relevance. However, for different queries, they apply the same model for documents from a specific language, which may be problematic as documents from this language may contribute different values for different queries (e.g., there are a lot of relevant documents in German for query A but very few for query B).

Based on this observation, we propose query-specific and language-specific results merging algorithms similar to those of federated search. For each query and each language, a few top ranked documents from each resource are downloaded, indexed and translated to English. Language-independent document scores are calculated for those downloaded documents and a logistic model is built for mapping all document scores in this ranked list to comparable language-independent document scores. Multiple logistic models are built in a similar manner for ranked lists in different languages and comparable scores can be estimated for all documents. Finally, all documents are ranked according to their comparable document scores. Experiments have been conducted to show that query-specific and language-specific merging algorithms outperform several other results merging algorithms. Furthermore, the query-specific and language-specific merging algorithms need to process (i.e., download, index and translate) very limited amount of documents (e.g., 10 per <query, language> pair) to acquire accurate results.

2. Multilingual Retrieval System

Accurate multilingual retrieval results are generated in this work by combining retrieval results from multiple multilingual retrieval methods. Specifically, we consider retrieval algorithms based on translating queries and retrieval algorithms based on translating documents. This section first describes basic text preprocessing procedures for different languages. Then it presents details of multilingual retrieval algorithms based on query translation and document translation, and finally proposes methods to combine the results from multiple multilingual retrieval algorithms.

2.1 Text Preprocessing

Stopword Lists: One of the first steps of preprocessing text documents for information retrieval is to throw away stopwords. The Inquery stopword list [3] is used in this work for English documents. Stopword lists of Finnish, French, German, Italian, Spanish and Swedish are acquired from[1], while the snowball stopword[2] list is used for Dutch.

Stemming: After stopwords have been excluded, other content words are stemmed by different stemming algorithms. Porter stemmer is used for English words. Dutch stemming algorithm is acquired from[2] and stemming algorithms from[1] are used for the other six languages.

Decompounding: Dutch, Finnish, German and Swedish are compound rich languages. All words that appear in the CLEF corpus and have lengths of more than 3 are considered as potential base words. In order to avoid too aggressive decompounding, we only consider base words that have higher collection frequencies than the word in consideration. Specifically, linking elements as –s-, -e-, and –en are

[1] http://www.unine.ch/info/clef/

[2] http://www.snowball.tartarus.org/

considered for Dutch, no linking elements for Finnish, elements as –s-, -n-, -e- and –en- for German and –s-, -e-, -u- and –o- for Swedish. The same set of decompounding procedure has been used in previous research [7].

Parallel corpus for word translation: Online machine translation [5,14,15] systems have been utilized to translate queries and documents for multilingual information retrieval systems. However, online systems may be updated, converted to commercial use and become unavailable [13]. If free connections still exist, the translation via online systems is associated with large amount of communication cost and can be very slow. Therefore, the translation process in this work is mainly accomplished in a word-by-word manner by using translation matrices generated by parallel corpus. Specifically, the parallel corpus of European Parliament proceedings 1996-2001[3] is used to build seven pairs of models between English and other seven languages. The GIZA++ [10] tool is utilized to build the mappings of translating English words into words of the other languages or translating words in other languages into English words. Each translation pair is associated with a probability value that indicates how probable the translation is.

2.2 Multilingual Retrieval via Query Translation

One straightforward multilingual retrieval method is to translate English queries into other languages, and then search those translated queries and merge the retrieval results from different languages into a single multilingual ranked list.

English query words are first translated into words in other languages by using translation matrices built from parallel corpus. Each English word is translated into top three candidates in the translation matrices of other languages. All the three translated words of an English word are associated with normalized weights (i.e., the sum of the weights is 1) according to the weights in translation matrices. One implicit problem of translating words with the help from parallel corpus is that some English words may not have translations as the vocabulary of the parallel corpus is limited. Therefore, we utilize word-by-word translation results from online machine translation software Systran[4] as a complement. As the number of words within the queries (total 60 queries) is limited, the communication cost of acquiring these translations from Systran is small. Specifically, all English queries terms are translated into words in six other languages except Dutch (Systran does not provide translation service from English to Dutch). These sets of translation representations are combined with translations built from parallel corpus while weight of 0.2 is assigned to the translation by Systran and weight of 0.8 is assigned to the translation with parallel corpus.

The translated queries are used to search indexes built in each language. Okapi [11,12] retrieval algorithm is applied to accomplish this and each query term is weighted by its weight in the translation representation. Bilingual retrieval results are acquired for those translated queries as well as monolingual results of English queries. As the same retrieval algorithm is applied on corpus of different languages with original/translated queries of the same lengths (i.e., the sum of weights of words in translated queries is always equal to the length of original English query), the raw scores in the ranked lists are somewhat comparable. Therefore, these ranked lists are merged together by their resource-specific scores into a final ranked list.

Another multilingual retrieval algorithm based on query translation takes advantage of query expansion by pseudo relevance feedback. Specifically, for resource of each language, query expansion is accomplished by adding 10 most common query terms within top 10 ranked documents of the initial retrieval result. The refined queries are used to generate new ranked lists of the resources and the ranked lists are then merged together.

2.3 Multilingual Retrieval via Document Translation

An alternative multilingual retrieval method is to translate all documents in other languages into English and apply the same original English queries. This method may have advantage against the

[3] http://people.csail.mit.edu/koehn/publications/europarl/
[4] http://www.systransoft.com/index.html

retrieval method based on query translation as the translation of longer documents may better represent the semantic meaning than the translation of short queries. Previous research [5] has also shown that the translation of a word from another language to English may be complementary to the translation from English to this language. For example, although one term in English may not be correctly translated into the corresponding German word, this German word may be correctly translated into the English term.

The document translation work is conducted using translation matrices built from parallel corpus. For each word in a language other than English, its top three English translations are considered. Five word slots are allocated to the three candidates of each untranslated word with proportion to the normalized translation probabilities of the three words. All the translated documents as well as the original English documents are collected into a single database and indexed.

Furthermore, the Okapi retrieval algorithm is applied on the single indexed database with original English queries to retrieve documents. Okapi retrieval algorithm without query expansion as well as Okapi retrieval algorithm with query expansion by pseudo relevance feedback (i.e., 10 additional query terms from top 10 ranked documents) is used in this work.

2.4 Combine Multilingual Ranked Lists

The basic assumption of improving ranking accuracy by combining ranked lists is that relevant documents are generally retrieved by multiple multilingual retrieval algorithms while different retrieval algorithms tend to retrieve different irrelevant documents. Similar idea has been successfully utilized in Metasearch of information retrieval [1].

Therefore, one simple combination algorithm is proposed to favor documents retrieved by more retrieval methods as well as high ranking documents retrieved by single types of retrieval methods. Let drs$_{k\_mj}$ denote the resource-specific raw document score for the jth document retrieved from the mth ranked list for kth query, drs$_{k\_m\_max}$ and drs$_{k\_m\_min}$ represents the maximum and minimum document scores in this ranked list respectively. Then, the normalized score of the jth document is calculated as:

$$d_{s_{k\_mj}} = \frac{(d_{rs_{k\_mj}} - d_{rs_{k\_m\_\min}})}{(d_{rs_{k\_m\_\max}} - d_{rs_{k\_m\_\min}})} \qquad (1)$$

where ds$_{k\_mj}$ is the normalized document score. After the normalization step, the document scores among all ranked lists are summed up for a specific document and all documents can be ranked accordingly. Note that this method can be seen as a variant of the well-known CombSUM [8] algorithm for Meta information retrieval. This method is called equal weight combination method in this work.

One particular issue about the proposed simple combination method is that it uses linear method (i.e., Equation 1) to normalize document scores and it treats the votes from multiple systems with equal weights. It is possible to design better score normalization method as well as more sophisticated weights for different systems in order to achieve better ranking accuracy. The idea is used in our algorithm to learn better score normalization method and the weights of systems with the help of training data. Formally, let us assume that there are M ranked lists to combine, and the normalized document scores of the mth ranked list are calculated as in Equation 1. Then the final combined document scores for a specific document d is calculated as:

$$score_{final}(d) = \frac{1}{M}\sum_{m=1}^{M} w_m score_m(d)^{r_m} \qquad (2)$$

where score$_{final}$(d) is the final combined document score and score$_m$(d) (which is zero if the document is not in the mth ranked list) represents the normalized score for this document from the mth ranked list. $\vec{w} = \{w_1,...,w_M\}$ and $\vec{r} = \{r_1,...,r_M\}$ are the model parameters, where the pair of (w$_m$, r$_m$) represents the weight of the vote and the exponential normalization factor for the mth ranked list respectively. The final ranked list can thus be generated with respect to the final scores calculated from Equation 2.

Maximizing ranking accuracy is the rule to derive desired model parameters of this combination model. In this work, ranking accuracy is represented formally as mean average precision (MAP) criterion. Let us assume that there are K training queries, the MAP criterion is represented formally as:

Language	**Dutch**	**English**	**Finnish**	**French**	**German**	**Italian**	**Spanish**	**Swedish**
All(MAP)	0.441	0.436	0.361	0.454	0.448	0.421	0.462	0.354

Table 1. Language-specific retrieval accuracy in mean average precision of retrieval results based on query translation with query term expansion.

Language	**Dutch**	**English**	**Finnish**	**French**	**German**	**Italian**	**Spanish**	**Swedish**
All(MAP)	0.386	0.460	0.434	0.418	0.442	0.415	0.439	0.357

Table 2. Language-specific retrieval accuracy in mean average precision of retrieval results based on document translation with query term expansion.

Methods	**Train**	**Test**	**All**
Qry_fb	0.317	0.353	0.341
Doc_nofb	0.346	0.360	0.356
Qry_nofb	0.312	0.335	0.327
Doc_fb	0.327	0.332	0.330
UniNe	0.322	0.330	0.327

Table 3. Mean average precision of multilingual retrieval methods. Qry means by query translation. Doc means by document translation, nofb means no pseudo relevance feedback, fb means pseudo relevant back.

Methods	**Train**	**Test**	**All**
M2_W1	0.384	0.431	0.416
M2_Trn	0.389	0.434	0.419
M3_W1	0.373	0.423	0.406
M3_Trn	0.383	0.431	0.415
M4_W1	0.382	0.432	0.415
M4_Trn	0.389	0.434	0.419
M5_W1	0.401	0.446	0.431
M5_Trn	0.421	0.449	0.440

Table 4. Mean average precision of merged multilingual list of different methods. M_X means to combine X results in the order of: 1). query translation with feedback, 2). document translation without feedback, 3). query translation without query expansion, 4). document translation with query expansion and 5). UniNE system. W1: means combine with equal weight, Trn means combine with trained weights.

$$\frac{1}{K}\sum_{k}\sum_{j\in D_k^+}\frac{rank_k^+(j)}{j} \tag{3}$$

where D_k^+ is the set of the ranks of relevant documents in the final ranked list for kth training query, and $rank_k^+(j)$ is the corresponding rank only among relevant documents. The multilingual retrieval task of CLEF as well as many other information retrieval evaluations uses the MAP criterion to evaluate retrieval accuracy.

In order to avoid the overfitting problem of model parameter estimation, two regularization items are introduced for $\vec{w}$ and $\vec{r}$ respectively. Together with the ranking accuracy criterion in Equation 3, the training optimization problem is represented as follows:

$$(\vec{w},\vec{r})^* = \arg\max_{\vec{w},\vec{r}}(\log\left(\frac{1}{K}\sum_{k}\sum_{j\in D_k^+}\frac{rank_k^+(j)}{j}\right)-\sum_{m=1}^{M}\frac{(w_m-1)^2}{2*a}-\sum_{m=1}^{M}\frac{(r_m-1)^2}{2*b}) \tag{4}$$

where $(\vec{w},\vec{r})^*$ is the estimated model parameters and (a,b) are two regularization factors that are set to 4 in this work. This problem is not a convex optimization problem and multiple local maximal values exist. A common solution is to search with multiple initial points.

After the desired model parameters have been estimated, it can be applied on test queries to combine ranked lists of different retrieval systems. This method is called learning combination method in this work.

3. Experimental Results: Multilingual Retrieval

Multilingual retrieval results are composed of documents from different languages. Therefore, it is helpful to first investigate the retrieval accuracies of results from single types of languages. Table 1 shows the bilingual retrieval results by translating English queries into other languages and the monolingual retrieval result of English. All the runs have utilized query term expansion by pseudo relevant feedback as described in Section 2. It can be seen that the retrieval accuracies of singe types of languages vary from 0.35 to 0.46. Table 2 shows the monolingual English retrieval result and bilingual retrieval results by translating documents in other languages into English and searching with English queries. Query expansion has been used for this retrieval method and the additional terms are extracted from the top 10 ranked documents among all English documents and the translated documents from other languages. It can be seen that although the retrieval method based on query expansion and the retrieval method based on document expansion generate similar results on several language (e.g., German and Spanish), their effectiveness on some other languages (e.g., Dutch and Finnish) are rather different. Note that the monolingual retrieval results of English with methods based on query translation and document translation are not exactly the same due to different configurations of retrieval methods (e.g., different query expansion methods).

The English monolingual retrieval results and bilingual retrieval results are merged together into a multilingual ranked list by the raw document scores. Table 3 shows the results of five multilingual retrieval algorithms on training queries (first 20 queries), test queries (next 40 queries) and the overall accuracy. It can be seen that multilingual retrieval algorithms based on query translation and algorithms based on document translation produce results of similar accuracy (i.e., 0.327-0.356), while the retrieval method based on document expansion that does not use query expansion has a small advantage. The results from multilingual retrieval system by [15] (merged by the trained logistic transformation model by maximizing MAP as described in Section 4.1) are also shown in Table 3 as it is considered in this work for multilingual result combination. It can be seen that the accuracy of UniNE system is very close to the other four algorithms. Table 3 suggests that ranked lists of individual multilingual retrieval systems may not be very effective compared to bilingual results (i.e., bilingual results in Table 1, 2 and 5).

One key step to improve the accuracy of multilingual retrieval result is to combine results of several multilingual retrieval methods. Two combination methods described in Section 2 as equal weight combination method and learning combination method are applied in this work. They are used to combine the results of the five retrieval algorithms described above. The combination results are shown in Table 4. It can be seen that the accuracies of combined multilingual result lists are substantially higher than the accuracies of results from single types of multilingual retrieval algorithms. This demonstrates the power to combine multilingual retrieval results. Detailed analysis shows that although the training combination method is consistently better than the equal weight combination method for the same configurations (i.e., the same number of ranked lists to combine), its advantage is very small. One possible reason is that the accuracies of the five retrieval algorithms are close and it does not make too much difference to adjust the voting weights among them.

4. Results Merge for Multilingual Retrieval

The second task we participated in CLEF 2005 is results merging for multilingual retrieval. Two sets of ranked lists across eight different languages are provided within this task and the goal is to merge these individual ranked lists together into two single lists with high accuracy. This is a difficult task as: i). Ranked lists from different languages may have different score ranges due to different retrieval strategies such as methods of query translation or query expansion [14,15]; ii). The corpus statistics (e.g., inverse document frequency) of different languages may be quite different; and iii). In multilingual federated search environment, there is no control over retrieval algorithms that the resources use. These characteristics make it hard to directly compare document scores among ranked lists of different languages.

Previous research [14,15] has proposed solution of learning query-independent and language-specific model by relevance judgment of previous queries to transform language-specific document scores into probabilities of relevance so that documents across different languages can be ranked by their estimated probabilities of relevance. However, this method may not be very accurate as a single query-independent transformation model is built for each language but the retrieved results of different queries of this language may have different characteristics. An alternative approach is to index all returned documents across different languages and apply a retrieval algorithm with the same retrieval strategy to compute comparable document scores. This method can be more accurate than the first approach as results from different queries are treated separately. However, this approach is associated with a large amount of computation costs and possible communication costs (i.e., when documents of different languages can only be accessed by sending requests to remote servers).

In this paper, a new approach is proposed to learn query-specific and language-specific models of translating language-specific document scores into comparable document scores. In particular, a small set of documents from each language is indexed at retrieval time to compute comparable document scores, and then a query-specific and language-specific model is trained by both comparable document scores and language-specific document scores of this small set of documents. By applying these models on ranked lists of all languages, comparable document scores can be obtained for all the returned documents and the final ranked list can be achieved. This approach has an advantage to avoid the requirement of human relevance judgment data for training. It only uses automatically computable document scores as surrogate of relevance judgment data and thus is similar as the semi-supervised learning results merging method in federated search [16]. Empirical study shows that this new approach is effective and high accuracy can be achieved by indexing a small number of documents at retrieval time.

This section is organized as follows: In section 4.1, an approach of learning query-independent and language-specific logistic transformation merging model is described and a new extension of learning the model by maximizing mean average precision is proposed; In Section 4.2, we describe the new approach of learning query-specific and language-specific result merging algorithm.

4.1 Learn Query-Independent and Language-Specific Merging Model via Relevance Training Data

To make the retrieved results from different ranked lists comparable, one natural idea is to map all the document scores into the probabilities of relevance and rank all documents accordingly. Particularly, logistic transformation model has been successfully utilized in previous study to achieve this goal [14,15]. This method has been shown to be more effective than round robin results merging, raw score results merging and several other alternatives. Let us assume that there are altogether I ranked lists from different languages to be merged, each of them provides J documents for each query and there are altogether K training queries with human relevance judgment. Particularly, d_{k_ij} represents the jth document from the ith language of training query k. The pair (r_{k_ij}, ds_{k_ij}) represents the rank of this document and the document score (normalized by Equation 1) respectively. By the logistic transformation model, the estimated probability of relevance of this document is:

$$P(rel \mid d_{k\_ij}) = \frac{1}{1+\exp(a_i r_{k\_ij} + b_i ds_{k\_ij} + c_i)} \tag{5}$$

where a_i ,b_i and c_i are the parameters of language-specific model that transforms all document scores of different queries from the ith language into the corresponding probabilities of relevance. The optimal parameter values are acquired generally by maximizing the log-likelihood (MLE) of training data, which is formally represented as:

$$\sum_{k,i,j} P^*(rel \mid d_{k\_ij}) \log(P(rel \mid d_{k\_ij})) \tag{6}$$

where P*(rel|d_{k_ij}) is the empirical probability value of a particular document. This is derived from human relevance judgment data, which is 1 when this document is relevant and 0 otherwise. This objective function is a convex function, which has only one global optimal solution.

One particular issue of training logistic transformation model by maximizing log-likelihood is that it equally treats each relevant document. However, this may not be a desired criterion in real world application. For example, a relevant document out of total 2 relevant documents for a query is generally more important to users than a relevant document out of total 100 relevant documents for another query. Therefore, queries are generally treated equally in information retrieval evaluation instead of individual relevant documents. This is formally represented by the mean average precision (MAP) criterion as described in Equation 3. The multilingual retrieval task of CLEF as well as many other information retrieval tasks uses the MAP criterion to evaluate retrieval accuracy.

One natural extension of training logistic transformation model by MLE criterion is to train the model with MAP criterion. Particularly, different sets of model parameters {a_i ,b_i and $c_{i,}$ 1<=i<=I} generate different sets of relevant documents as $\{D_k^+, 1 <= k <= K\}$ and thus achieve different MAP values. The training procedure of maximizing MAP searches for a set of model parameters that generates the highest MAP value. However, this problem is not a convex optimization problem and multiple local maximal values exist. A common solution is to search with multiple initial points.

The new algorithm of training logistic model for mean average precision is called logistic model with MAP goal in this paper. This is believed to be a more direct method to improve the MAP value of multilingual retrieval system in CLEF and also a better method to reflect users' preference than training logistic model with MLE criterion.

4.2 Learn Query-Specific and Language-Specific Merging Model

One particular problem about language-specific logistic transform merging model introduced in Section 4.1 is that it applies the same model on results of different queries from each language. This is problematic when result lists of different queries have similar score distributions but have different distributions of probability of relevance. This suggests that query-specific model should be studied for high merging accuracy of multilingual retrieval.

Previous research has proposed query-specific merging method that uses the two step Retrieval Status Values (RSV) [9,13]. For each query, this method indexes top ranked documents of different languages at the retrieval time and computes comparable document scores. One choice is to translate all top ranked documents (i.e., 1000) of different languages into a single language, index them and apply a singe centralized retrieval algorithm to generate a final ranked list. However, this method is associated with a large amount of computation costs of translating and indexing many documents.

In multilingual federated search environment, the cost of processing retrieved documents is even higher as the contents of all documents to translate are not directly accessible and they must be downloaded from corresponding servers. This is also true for a multilingual federated search environment where contents of all available documents cannot be directly crawled into a single centralized database. This means that generally the corpus statistics (e.g., corpus inverse document frequencies) are not available and can only be simulated by collecting statistics from sampled documents.

To propose methods that work in stricter environments, we follow the multilingual federated search approach in this research. There exists a resource that contains all documents from one language. These resources provide searching services of their documents.

Query-based sampling method is utilized in this work to learn corpus statistics from each resource with a particular language [4]. Specifically, random one-term queries are sent to each resource and retrieve about 4 documents for each query. Altogether 3,000 documents are collected from each resource. The sampled documents from each resource are collected together to create the centralized sample database for this resource so that corpus statistics such as inverted document frequencies can be estimated from this database.

The above paragraphs describe the procedures to acquire estimated corpus statistics in multilingual federated search environment. With this information, retrieved documents from individual resources will be assigned comparable document scores and merged into a single final ranked list. Previous research [5] and the empirical results of two-years-on multilingual retrieval task in this work demonstrate the advantage of utilizing evidence by both translating queries and translating documents. The goal of the query-specific and language-specific results merging algorithms in this work is to assign comparable document scores to all retrieved documents by combining document scores of retrieval methods based on query translation and scores based on document translation.

To obtain comparable document scores based on query translation, the original English query is first translated into other languages in word-by-word manner by using translation matrices as described in Section 2. These translated queries and the original English query are sent to the eight resources and retrieve eight sets of individual ranked lists. As in federated search environment, it is generally difficult to require all resources to use the same type of retrieval algorithm with the same type of configuration (e.g., Okapi with the same feedback procedure). The returned document scores may not be directly comparable. Therefore, to compute comparable scores of retrieved documents, the documents need to be downloaded and indexed, and then the same retrieval algorithm is applied on the downloaded documents with the same configuration. Particularly, the retrieved documents are downloaded and an

Okapi retrieval algorithm is applied on these documents with corpus statistics from the centralized sample database of the corresponding resource.

Comparable document scores based on document translation are acquired by applying a single Okapi retrieval method on all retrieved English documents and all the translated documents from resources with other languages. Specifically, all retrieved documents in languages other than English are first translated into English in word-by-word manner using translation matrices, and then are merged into a single set of documents with documents that are originally in English. Furthermore, this set of documents is indexed and an Okapi retrieval algorithm is applied on this set of documents with corpus statistic from the centralized sample database of English resource. As this results merging method downloads (also indexes and translates) all documents in the given ranked lists, it is called complete downloading method.

Two sets of comparable document scores based on retrieval methods of query translation and document translation are merged together into a single set with the method described in Section 2. Specifically, the two sets of scores are first normalized separately and then are combined into a new ranked list with the equal weight combination method described in Section 2.

It can be noted from the description that a large amount of online costs is associated with the complete downloading result merging algorithm. Within federated search environment, communication cost is associated with downloading each document. Furthermore, computation costs as indexing and translation are also associated to process each downloaded document. These problems are particularly serious as they happen in an online manner. Therefore, a more efficient algorithm is much more desired for operational system.

The key idea to calculate comparable document scores more efficiently is to only calculate scores for a small set of representative documents. Particularly, a small set of retrieved documents from each resource is selected; the above procedure of downloading and calculating new scores based on query translation and document translation is applied on this set of documents. These documents that have both language-specific scores and calculated comparable scores serve as training data for learning a logistic model, which estimates the comparable document scores for other documents that have not been downloaded and indexed.

Generally top ranked documents of retrieved documents from each resource are more probable to be relevant, they are selected for downloading and calculating comparable document scores. Let us assume top L documents from the ranked list of each resource are downloaded to calculate comparable scores. Let the pair (dc$_{k'\_il}$, ds$_{k'\_il}$) denote the normalized comparable document score and normalized language-specific score for the lth downloaded document of the ith resource for k' the query. Let the pair (a$_{k'\_i}$, b$_{k'\_i}$) denote the parameters of the corresponding query-specific and language-specific model. These parameters are learned by solving the following optimization problem to minimize the mean squared error between exact normalized comparable scores and the estimated comparable scores as:

$$\left(a^*_{k'\_i}, b^*_{k'\_i}\right) = \underset{(a,b)}{\arg\min} \sum_{d_{k'\_il} \in D_L \vee D_{NL}} \left(d_{C_{k'\_il}} - \frac{1}{1+\exp(a * d_{S_{k'\_il}} + b * 1)}\right)^2 \qquad (7)$$

where D$_L$ is the downloaded L documents from the resource and D$_{NL}$ is a pseudo set of L documents with pseudo normalized comparable scores zero and pseudo normalized language-specific scores zero. This set of pseudo documents is introduced in order to make sure that the learned model ranks documents in the correct way (i.e., documents with higher language-specific scores are ranked higher in the ranked list with comparable scores than documents with lower language-specific scores).

Finally, logistic models can be learned for all resources in the same way. They are applied to all retrieved documents from all resources and the documents can be ranked according to their estimated comparable scores. Note that only language-specific document scores are used in the logistic model in Equation 7 while document ranks in language-specific ranked lists are not considered. This is different from Equation 5, and is used here in order to reduce the number of parameters for the limited amount of data (i.e., the small set of documents with both comparable scores and language-specific scores).

Note that exact comparable document scores are available for the documents that have been downloaded and processed. One method to take advantage of these scores is to combine them with the estimated scores. In this work, they are combined together with equal weights (i.e., 0.5).

Language	**Dutch**	**English**	**Finnish**	**French**	**German**	**Italian**	**Spanish**	**Swedish**
All(MAP)	0.431	0.536	0.192	0.491	0.513	0.486	0.483	0.435

Table 5. Language-specific retrieval accuracy in mean average precision of retrieval results from UniNE system.

Language	**Dutch**	**English**	**Finnish**	**French**	**German**	**Italian**	**Spanish**	**Swedish**
All(MAP)	0.236	0.514	0.163	0.350	0.263	0.325	0.298	0.269

Table 6. Language-specific retrieval accuracy in mean average precision of retrieval results from HummingBird system

Methods	**Train**	**Test**	**All**
TrainLog_MLE	0.301	0.301	0.301
TrainLog_MAP	0.322	0.330	0.327

Table 7. Mean average precision of merged multilingual lists of different methods on UniNE result lists. TrainLog_MLE means trained logistic transformation model by maximizing MLE. TrainLog_MAP means trained logistic transformation model by maximizing MAP.

Methods	**Train**	**Test**	**All**
TrainLog_MLE	0.186	0.171	0.176
TrainLog_MAP	0.210	0.192	0.198

Table 8. Mean average precision of merged multilingual lists of different methods on HummingBird result lists. TrainLog_MLE means trained logistic transformation model by maximizing MLE. TrainLog_MAP means trained logistic transformation model by maximizing MAP.

5. Experimental Results: Results Merge

This section presents the experiment results of different results merging algorithms for ranked lists of different languages. These results merging algorithms work on two sets of ranked lists from UniNE system [14,15] and HummingBird system[5]. Both of the two sets are composed of ranked lists from eight different languages. The language-specific retrieval accuracies of ranked lists of UniNE and HummingBird systems are shown in Table 5 and Table 6 respectively. It can be seen from Table 5 that all language-specific ranked lists generated by UniNE system except ranked list of Finnish have high accuracy. On the other side, the accuracies of ranked lists generated by HummingBird system are much lower than those of the UniNE system. These two sets of ranked lists are good candidates to evaluate the effectiveness of merging algorithms for both accurate ranked lists and inaccurate ranked lists.

The first two results merging algorithms to evaluate are the query-independent and language-specific results merging algorithms by optimizing the maximum likelihood criterion (MLE) and the mean average precision (MAP) criterion respectively. Their merging accuracies on both the ranked lists of UniNE system and HummingBird systems are shown in Table 7 and Table 8. It can be seen that merging accuracies of the UniNE system is much higher than those of the HummingBird system. This is consistent with our expectation as the language-specific ranked lists of UniNE system are better than those of HummingBird system. The merging accuracies of learning algorithms on UniNE system are similar to those reported in [15]. Furthermore, it can be seen from both Tables 7 and 8 that the learning algorithm optimized for mean average precision criterion is always more accurate than that optimized for maximum likelihood criterion. This demonstrates the power to directly optimize for mean average precision accuracy as treating different queries equally against the strategy of optimizing for maximum likelihood that does not directly evaluate mean average precision. However, the merging accuracy is not good compared to bilingual runs.

To improve the merging accuracy, query-specific and language-specific algorithms are introduced. Two types of algorithms are evaluated in this work. The first method downloads all documents from ranked lists of different languages and calculates comparable document scores (C_X). The second method downloads top ranked documents and calculates their comparable documents to build logistic models. These models generate estimated comparable document scores and finally combine the estimated scores with acquired exact comparable scores wherever they are available (Top_X_C05). The experimental results of different variants of these algorithms on UniNE system and HummingBird system are shown in Tables 9 and 10 respectively. Note that both these two algorithms do not require human relevance judgment for training data. Therefore, the results on training query set and test query set are obtained separately without using any relevance judgment data.

It can be seen from Table 9 and Table 10 that both these two query-specific and language-specific merging algorithms substantially outperform query-independent and language-specific algorithms. The accuracies of the two query-specific and language-specific methods (i.e., C_X and Top_X_C05) are close on the UniNE system. It is interesting that the Top_150_C05 method outperforms all C_X runs

[5] http://www.hummingbird.com/products/searchserver/

Methods	**Train**	**Test**	**All**
Top_150_C05	0.360	0.412	0.395
Top_30_C05	0.357	0.399	0.385
Top_15_C05	0.346	0.402	0.383
Top_10_C05	0.330	0.393	0.372
Top_5_C05	0.296	0.372	0.347
C_500	0.356	0.384	0.374
C_150	0.352	0.391	0.378
C_1000	0.356	0.382	0.373

Table 9. Mean average precision of merged multilingual lists of different methods on UniNE result lists. Top_x indicates x top documents are downloaded to generate logistic transformation model, C05 indicates both scores from logistic transformation model and centralized document scores are utilized when they are available and they are combined with a linear weight as 0.5. C_X means top X documents from each list are merged by their centralized doc scores.

Methods	**Train**	**Test**	**All**
Top_150_C05	0.278	0.297	0.291
Top_30_C05	0.260	0.268	0.265
Top_15_C05	0.235	0.253	0.247
Top_10_C05	0.222	0.248	0.239
Top_5_C05	0.210	0.234	0.226
C_500	0.315	0.333	0.326
C_150	0.290	0.302	0.298
C_1000	0.324	0.343	0.337

Table 10. Mean average precision of merged multilingual lists of different methods on HummingBird result lists. Top_x indicates x top documents are downloaded to generate logistic transformation model, C05 indicates both scores from logistic transformation model and centralized document scores are utilized when they are available and they are combined with a linear weight as 0.5. C_X means top X documents from each list are merged by their centralized doc scores.

on the UniNE system. This means that the combination of estimated comparable scores and exact comparable scores can be more accurate than exact comparable scores in some cases. Detailed analysis shows that the estimation of comparable document scores is related with document scores from ranked lists of UniNE systems. The estimated document scores can be seen as combination results from not only the two retrieval methods that based on query translation and document translation but also the retrieval method of UniNE system. Therefore, the combined results that are related with three retrieval systems may be better than those of exact comparable scores from two retrieval systems. It is encouraging to see that with very limited amount of downloaded documents, the Top_10_C05 method still has more than 10 percent advantage over the query-independent and language-specific result merging algorithms.

It can be seen from Table 10 that the advantage of query-specific and language-specific algorithms over query-independent and language-specific algorithms is even larger for the results on HummingBird system than those on UniNE system. This demonstrates the power of query-specific and language-specific merging algorithms for ineffective ranked lists. It is interesting to note that the Top_X_C05 runs are not as effective as C_X runs on HummingBird System. The reason can be explained that the ranked lists of HummingBird system are not as accurate as those of UniNE systems. The influence of document scores within ranked lists of HummingBird on the estimated comparable score is not as helpful as that from the UniNE system.

6. Conclusion:

This paper describes the algorithms we have studied and proposed for the CLEF 2005 evaluation tasks as: Multi-8 two-years-on retrieval task and Multi-8 results merging task.

For multi-8 two-years-on retrieval task, our focus is to generate and combine multilingual retrieval results that are built from simple bilingual (or monolingual) ranked lists. Specifically, we first generate multiple multilingual retrieval results by merging bilingual (or monolingual) retrieval results of same types of retrieval algorithms, and then combine the multilingual retrieval results together. Several combination methods have been proposed and empirical studies have demonstrated that the combination of multilingual retrieval results can substantially improve the accuracies over single multilingual ranked lists.

The task of Multi-8 results merging task is to merge two sets of eight bilingual (or monolingual for English) ranked lists into multilingual ranked lists. This is the primary interest of our work and we have proposed to apply results merging algorithm of federated search task for this problem. Top ranked documents within each ranked list are indexed and translated to compute comparable document scores. Query-specific and language-specific logistic models are built based on comparable document scores of these documents and also the scores of these documents in language-specific ranked lists. These logistic models have been built to estimate comparable document scores for all documents in ranked lists of different languages, and finally all documents are sorted accordingly. Experiments have shown that the new proposed methods outperform previous research and they only need to process (i.e., download, index and translate) very small amount of documents (e.g., 10 per <query, language> pair) to acquire accurate results.

Although query-specific and language-specific merging algorithm algorithms are much better than previous merging methods, in some cases their accuracies are still not at bilingual levels (e.g., UniNE systems). This suggests the necessity of more sophisticated result merging algorithms for future research.

7. Acknowledgement

Research presented in this paper has been supported in part by an ARDA grant under Phase II of the AQUAINT program. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor.

8. Reference:

[1]. Aslam, A. and Montague, M. 2001. Models for Metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[2]. Brown, P.F, Pietra, D., Pietra, D, Mercer, R.L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19: 263-312.

[3]. Callan, J., Croft W. B. and Broglio, J. 1995. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3).

[4]. Callan, J. and Connell, M. 2001. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2), pp. 97-130.

[5]. Chen, A. and F. C. Gey. 2003. Cross-language Retrieval Experiments at CLEF-2003. In C. Peters(Ed.), *Results of the CLEF2002 cross-language evaluation forum*.

[6]. Jones, G. J. F., Burke, M., Judge, J., Khasin, A., Lam-Adesina, A., Wagner, J. 2004. Dublin City University at CLEF 2004: Experiments in Monolingual, Bilingual and Multilingual Retrieval. In C. Peters(Ed.), *Results of the CLEF2004 cross-language evaluation forum*.

[7]. Kamps, J., Monz, C., Rijke, Maarten de. and Sigurbjörnsson, Börkur. 2003. The University of Amsterdam at CLEF 2003. In C. Peters(Ed.), *Results of the CLEF2003 cross-language evaluation forum.*

[8]. Lee. J. H. 1997. Analyses of multiple evidence combination. In *Proceedings of the 20th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval.*

[9]. Martinez-Santiago, Martin M. and Urena, A. 2002. SINAI on CLEF 2002: Experiments with merging strategies. In C. Peters(Ed.), *Results of the CLEF2002 cross-language evaluation forum*.

[10]. Och, F. J. and Hermann N. 2000. Improved Statistical Alignment Models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 440-447.

[11]. Ogilvie, P and Callan, J. 2001. Experiments using the Lemur toolkit. In *Proceedings of the Tenth Text Retrieval Conference (TREC-10).*

[12]. Robertson S. and Walker. S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[13]. Rogati. M. and Yang Y. M. 2003. CONTROL: CLEF-2003 with Open, Transparent Resources Off-Line. Experiments with merging strategies. In C. Peters(Ed.), *Results of the CLEF2003. cross-language evaluation forum*.

[14]. Savoy, J. 2002. Report on CLEF-2002 Experiments: Combining multiple sources of evidence. In C. Peters(Ed.), *Results of the CLEF2002 cross-language evaluation forum.*

[15]. Savoy, J. 2003. Report on CLEF-2003 Experiments. In C. Peters(Ed.), *Results of the CLEF2003 cross-language evaluation forum.*

[16]. Si, L. and Callan, J. 2003. "A Semi-Supervised Learning Method to Merge Search Engine Results" In *ACM Transactions on Information Systems*, 24(4). pp. 457-491.