Information Retrieval, 7, 149–182, 2004

Multilingual Information Retrieval Using Machine Translation, Relevance Feedback and Decompounding

AITAO CHEN aitao@sims.berkeley.edu
School of Information Management and Systems, University of California at Berkeley, CA 94720-4600, USA

FREDRIC C. GEY gey@ucdata.berkeley.edu
UC Data Archive & Technical Assistance (UC DATA), University of California at Berkeley, CA 94720-5100, USA

Received December 6, 2002; Revised July 18, 2003; Accepted July 21, 2003

Abstract. Multilingual retrieval (querying of multiple document collections each in a different language) can be achieved by combining several individual techniques which enhance retrieval: machine translation to cross the language barrier, relevance feedback to add words to the initial query, decompounding for languages with complex term structure, and data fusion to combine monolingual retrieval results from different languages. Using the CLEF 2001 and CLEF 2002 topics and document collections, this paper evaluates these techniques within the context of a monolingual document ranking formula based upon logistic regression. Each individual technique yields improved performance over runs which do not utilize that technique. Moreover the techniques are complementary, in that combining the best techniques outperforms individual technique performance. An approximate but fast document translation using bilingual wordlists created from machine translation systems is presented and evaluated. The fast document translation is as effective as query translation in multilingual retrieval. Furthermore, when fast document translation is combined with query translation in multilingual retrieval, the performance is significantly better than that of query translation or fast document translation.

Keywords: multilingual information retrieval, cross-language information retrieval, relevance feedback, decompounding, results merging

1. Introduction

Multilingual information retrieval (MLIR) is the task of searching for relevant documents in a collection of documents in more than one language in response to a query, and presenting a unified ranked list of documents regardless of language. Multilingual retrieval is an extension of bilingual retrieval where the collection consists of documents in a single language that is different from the query language. Recent developments on multilingual retrieval were reported in CLEF 2000 (Peters 2001), CLEF 2001 (Peters et al. 2002a), and CLEF 2002 (Peters 2002b). Multilingual retrieval methods fall generally into one of three groups. The first approach translates the source topics into all the document languages in the document collection. Then monolingual retrieval is carried out separately for each document language, resulting in one ranked list of documents for each document language. Finally the intermediate ranked lists of retrieved documents, one for each language, are merged to

yield a combined ranked list of documents regardless of language. Some examples of taking the first approach to MLIR are (Chen 2002a, Hiemstra et al. 2001, Savoy 2002a, McNamee and Mayfield 2002). The second approach translates a multilingual document collection into the topic language. Then the topics are used to search against the translated document collection. See for example (Braschler et al. 2002). The third approach also translates topics to all document languages, as in the first approach, but then the source topics (queries) and the translated topics are concatenated to form a set of multilingual topics. The multilingual topics are then searched against the multilingual document collection (where all documents are collected into a single index) to produce a ranked list of documents in all languages. The third approach to MLIR is taken by Gey et al. (2001). The latter two approaches do not involve merging two or more ranked lists of documents, one for each document language, to form a combined ranked list of documents in all document languages. The third approach is appealing in that it bypasses document translation, and circumvents the difficult merging problem. However, there is some empirical evidence showing that the third approach is less effective than the first one (Chen 2002a).

We believe that three of the core components of the first approach to multilingual retrieval are robust monolingual retrieval, topic translation, and merging. Performing multilingual retrieval requires the deployment of multiple language resources such as stopwords, stemmers, bilingual dictionaries, machine translation systems, parallel or comparable corpora. The end performance of multilingual retrieval can be affected by many factors, such as monolingual retrieval performance of the document ranking algorithm, the quality and coverage of the translation resources, the availability of language-dependent stemmers and stopwords, and the effectiveness of merging algorithms.

This paper evaluates the effectiveness of different techniques to multilingual information retrieval using only machine translation (MT) systems and investigates the impact of blind relevance feedback and decompounding on the performance of multilingual information retrieval. Alternative approaches to translation in multilingual retrieval include using manually constructed bilingual dictionaries which enumerate possible translations for each word in the topic language (see Ballesteros and Croft 1998), and aligned parallel texts at the document to sentence level to induce bilingual tables of probable translations (Yang et al. 1998). Xu and Weischedel (2001, 2002), among others, have done this on a large scale for Chinese and Arabic. Comparable corpora, such as news stories about the same subject or event written independently in each language and aligned temporally may also be used to create noisy bilingual lexicons (see Picci and Peters 1998). Pirkola et al. (2001) summarizes the dictionary-based approach to cross-language information retrieval.

For both cross-language retrieval and multilingual retrieval, it is more common to translate topics or queries into the document languages than to translate documents into the topic language, partly because it is faster and takes less effort to do so. However, when queries are short or not well-formed or incomplete sentences, the MT-based query translations may not be optimal. The study by Oard (1998) on comparing MT-based query translation and MT-based document translation cross-language approaches shows MT-based document translation outperforms MT-based query translation. Braschler et al. (2002) also compare the MT-based query translation and MT-based document translation using CLEF 2001 test collection.

In this paper we present an approximate but fast approach to translating a document collection into the query language in two steps. First, we collect all the unique words in the document collection, and then translate the unique words into the query language using a machine translation system. Second, we translate the documents word-by-word into the query language using the bilingual wordlist created in the first step. This approach is very efficient.

The paper is organized as follows. In Section 2 we briefly describe the document ranking algorithm based on logistic regression analysis. In Section 3 we describe a blind relevance feedback for the logistic regression-based document ranking algorithm. In Section 4 we describe a decompounding procedure for splitting a compound into its component words. In Section 5 we present an approximate but very fast document translation method. The test collections are briefly described in Section 6, and the topics and documents indexing procedure is presented in Section 7. Section 8 evaluates multiple merging strategies from separate monolingual rankings and compares them to unified indexes for both queries and documents.

2. Document ranking

A typical text retrieval system ranks documents according to their relevances to a given query. The documents that are more likely to be relevant are ranked higher than those that are less likely. In this section we briefly describe a logistic regression-based document ranking algorithm developed at Berkeley (Cooper et al. 1994). We used this document ranking algorithm for all the the retrieval runs reported in this paper. The log-odds (or the logit transformation) of the probability that document D is relevant with respect to query Q, denoted by $\log O(R \mid D, Q)$, is given by

$$\begin{aligned}\log O(R \mid D, Q) &= \log \frac{p(R \mid D, Q)}{1 - p(R \mid D, Q)} = \log \frac{p(R \mid D, Q)}{p(\bar{R} \mid D, Q)} \\ &= -3.51 + 37.4 * X_1 + 0.330 * X_2 - 0.1937 * X_3 + 0.0929 * X_4\end{aligned}$$

where D denotes a document and Q a query, R is a relevance variable, $p(R \mid D, Q)$ is the probability that document D is relevant to query Q, $p(\bar{R} \mid D, Q)$ the probability that document D is not relevant to query Q, which is $1.0 - p(R \mid D, Q)$. The four explanatory variables X_1, X_2, X_3, and X_4 are defined as follows:

$$X_1 = \frac{1}{\sqrt{M} + 1} \sum_{i=1}^{M} \frac{qtf_i}{ql + 35}$$

$$X_2 = \frac{1}{\sqrt{M} + 1} \sum_{i=1}^{M} \log \frac{dtf_i}{dl + 80}$$

$$X_3 = \frac{1}{\sqrt{M} + 1} \sum_{i=1}^{M} \log \frac{ctf_i}{cl}$$

$$X_4 = M$$

where M is the number of matching terms between a document and a query, qtf_i is the within-query frequency of the ith matching term, dtf_i is the within-document frequency of the ith matching term, ctf_i is the occurrence frequency in a collection of the ith matching term, ql is query length (i.e., number of terms in a query), dl is document length (i.e., number of terms in a document), and cl is collection length (i.e., number of terms in a test collection). If stopwords are removed from indexing, then ql, dl, and cl are the query length, document length, and collection length, respectively, after removing stopwords. If the query terms are re-weighted, then qtf_i is no longer the original term frequency, but the new weight, and ql is the sum of the new weight values for the query terms. Note that, unlike X_2 and X_3, the variable X_1 sums the "optimized" relative frequency without first taking the log over the matching terms. The relevance probability of document D with respect to query Q can be written as follows, given the log-odds of relevance probability.

$$p(R \mid D, Q) = \frac{1}{1 + e^{-\log O(R \mid D, Q)}}.$$

The documents are ranked in decreasing order by their relevance probability $p(R \mid D, Q)$ with respect to a query. The coefficients were determined by fitting the logistic regression model specified in $\log O(R \mid D, Q)$ to training data using a statistical software package. We refer readers to Cooper et al. (1994) for more details.

3. Relevance feedback

It is well known that blind (also called pseudo) relevance feedback can substantially improve retrieval effectiveness. It is commonly implemented in research text retrieval systems. See for example the papers of the groups who participated in the Ad Hoc tasks in TREC-7 (Voorhees and Harman 1998) and TREC-8 (Voorhees and Harman 1999). Blind relevance feedback is typically performed in two stages. First, an initial search using the original queries is performed, after which a number of terms are selected from the top-ranked documents that are presumed relevant. The selected terms are weighted and then merged with the initial query to formulate a new query. Finally the new query is searched against the document collection to produce a final ranked list of documents. Some of the issues involved in implementing blind relevance feedback include determining the number of top ranked documents that will be presumed relevant and from which new terms will be extracted, ranking the selected terms and determining the number of terms that should be selected, and assigning weights to the selected terms. The techniques for deciding the number of terms to be selected, the number of top-ranked documents from which to extract terms, and ranking the terms vary. We refer readers to the paper (Harman 1992) for a survey of relevance feedback techniques. In cross-language retrieval, query expansion can be carried out before query translation or after query translation or both. Ballesteros and Croft (1997) experimented with both pre-translation and post-translation query expansion in English to Spanish cross-language retrieval using a machine-readable dictionary as the translation resource. They found that both pre-translation and post-translation query expansion are helpful and combining them is even more effective. In this paper we only utilize post-translation relevance feedback.

The Berkeley document ranking formula has been in use for many years without blind relevance feedback. Chen (2002a) recently presented a technique for incorporating blind relevance feedback into the logistic regression-based document ranking framework. Two factors are import in relevance feedback. The first one is how to select the terms from top-ranked documents after the initial search, the second is how to assign weights to the selected terms with respect to the terms in the initial query.

3.1. Term selection methods

In this section we present five ways for selecting terms from the top-ranked documents that are presumed relevant after the initial search. The selection methods are

1. relevance weighting (RW),
2. mutual information (MI),
3. chi-square statistic (CHI),
4. likelihood ratio for multinomial distribution (LRM), and
5. relative frequency ratio (RFR).

The desired terms for query expansion are the ones that occur frequently in the documents that are presumed relevant, but infrequently in the remaining documents. To help better understand the first four term selection methods, we present a contingency table for a word below.

	(presumed) Relevant (R)	(presumed) Irrelevant ($\bar{R}$)	
Indexed (t)	n_1	n_2	n_5
Not indexed ($\bar{t}$)	n_3	n_4	n_6
	n_7	n_8	n

where $n = n_1 + n_2 + n_3 + n_4 = n_5 + n_6 = n_7 + n_8$, $n_5 = n_1 + n_2$, $n_6 = n_3 + n_4$, $n_7 = n_1 + n_3$, $n_8 = n_2 + n_4$. n is the number of documents in the collection, n_7 the number of top-ranked documents after the initial search that are presumed relevant, n_1 the number of documents among the n_7 top-ranked documents that contain the term t, and n_5 the number of documents in the collection that contain the term t.

3.1.1. Relevance weighting. For term selection, we assume some top-ranked documents after the initial search are relevant, and the rest of the documents in the collection are irrelevant. For the terms in the documents that are presumed relevant, we compute the ratio of the odds of seeing a term in a relevant document over the odds of seeing the same term in an irrelevant document. This is the term relevance weighting formula proposed by Robertson and Sparck Jones (1976). From the word contingency table we see that that the probability of finding a term t in a relevant document is $p(t \mid R) = \frac{n_1}{n_7}$, because n_1 documents out of n_7 relevant documents contain the term t. Likewise, the probability of not finding

the term t in a relevant document is $p(\bar{t} \mid R) = \frac{n_3}{n_7}$, because n_3 documents out of n_7 relevant documents do not contain the term t. The odds of finding a term t in a relevant document is $O(t \mid R) = \frac{p(t \mid R)}{1-p(t \mid R)} = \frac{p(t \mid R)}{p(\bar{t} \mid R)} = \frac{n_1/n_7}{n_3/n_7} = \frac{n_1}{n_3}$. Likewise, the odds of finding a term t in an irrelevant document is $O(t \mid \bar{R}) = \frac{p(t \mid \bar{R})}{1-p(t \mid \bar{R})} = \frac{p(t \mid \bar{R})}{p(\bar{t} \mid \bar{R})} = \frac{n_2/n_8}{n_4/n_8} = \frac{n_2}{n_4}$. The terms extracted from the n_7 top-ranked documents are ranked by the log of odds-ratio which is given by

$$w_t = \log \frac{O(t \mid R)}{O(t \mid \bar{R})} = \log \frac{\frac{n_1}{n_3}}{\frac{n_2}{n_4}} = \log \frac{n_1 n_4}{n_2 n_3}. \tag{1}$$

Robertson et al. (2000) used a similar method for term selection. Their method can be expressed as $n_1 * w_t$ in our notation.

3.1.2. Mutual information. In this section we present an alternative statistic for ranking and selecting terms from the top-ranked documents that are presumed relevant. This statistic is the mutual information between the event that a randomly selected document is relevant and the event that a randomly selected document contains a given term in the document. The mutual information, $MI(R, t)$, is given by

$$MI(R, t) = \log \frac{p(R, t)}{p(R) * p(t)} = \log \frac{\frac{n_1}{n}}{\frac{n_7}{n}\frac{n_5}{n}} = \log \frac{n_1 n}{n_5 n_7}$$

where $p(R, t)$ is the probability that a randomly selected document from the collection is relevant *and* contains the term t, $p(R)$ is the probability that a randomly selected document is relevant, and $p(t)$ is the probability that a randomly selected document contains the term t. $p(R, t)$ can be estimated by $\frac{n_1}{n}$, since n_1 documents out of n documents are relevant *and* contain the term t. Similarly, $p(R)$ can be estimated by $\frac{n_7}{n}$, and $p(t)$ by $\frac{n_5}{n}$.

3.1.3. Chi-square statistic. Chi-square statistic is computed as

$$\chi^2 = \sum_{i=1}^{4} \frac{(n_i - e_i)^2}{e_i}$$

where n_i is the observed count of the ith cell in the word contingency table, and e_i is the expected count of the ith cell under the assumptions that the relevance or irrelevance of a randomly selected document is independent of the presence or absence of a term. We define four probabilities, one for each cell in the word contingency table, as follows:

$$\begin{aligned} p_1 &= p(R, t), \\ p_2 &= p(\bar{R}, t), \\ p_3 &= p(R, \bar{t}), \\ p_4 &= p(\bar{R}, \bar{t}) \end{aligned}$$

where p_1 is the probability that a randomly selected document D is relevant and contains the term t, p_2 the probability that a randomly selected document D is irrelevant and contains

the term t, p_3 the probability that a randomly selected document D is relevant but does not contain the term t, and p_4 the probability that a randomly selected document D is irrelevant and does not contain the term t. If we assume that the presence or absence of a term in a randomly selected document is independent of the relevance or irrelevance of the document, then we can estimate the probabilities p_1 through p_4 as follows:

$$\bar{p}_1 = p(R, t) = p(R) * p(t) = \frac{n_7}{n} * \frac{n_5}{n} \tag{2}$$

$$\bar{p}_2 = p(\bar{R}, t) = p(\bar{R}) * p(t) = \frac{n_8}{n} * \frac{n_5}{n} \tag{3}$$

$$\bar{p}_3 = p(R, \bar{t}) = p(R) * p(\bar{t}) = \frac{n_7}{n} * \frac{n_6}{n} \tag{4}$$

$$\bar{p}_4 = p(\bar{R}, \bar{t}) = p(\bar{R}) * p(\bar{t}) = \frac{n_8}{n} * \frac{n_6}{n}. \tag{5}$$

Under the independence assumptions, the expected counts can be computed as follows:

$$e_1 = n * \bar{p}_1 = \frac{n_5 n_7}{n}$$
$$e_2 = n * \bar{p}_2 = \frac{n_5 n_8}{n}$$
$$e_3 = n * \bar{p}_3 = \frac{n_6 n_7}{n}$$
$$e_4 = n * \bar{p}_4 = \frac{n_6 n_8}{n}.$$

Now the chi-square statistic for term t can be computed as follows:

$$\chi^2 = \frac{(n_1 - e_1)^2}{e_1} + \frac{(n_2 - e_2)^2}{e_2} + \frac{(n_3 - e_3)^2}{e_3} + \frac{(n_4 - e_4)^2}{e_4}.$$

The terms in the top-ranked documents that are presumed relevant are ranked in descending order by their Chi-square values.

3.1.4. Likelihood ratio for multinomial distribution. Here we assume the counts for the four cells in the word contingency table follow a multinomial distribution with density function

$$f(n_1, n_2, n_3, n_4 \mid p_1, p_2, p_3, p_4) = \frac{n!}{n_1! n_2! n_3! n_4!} p_1^{n_1} p_2^{n_2} p_3^{n_3} p_4^{n_4}.$$

The maximum likelihood estimate of p_i is $\hat{p}_i = \frac{n_i}{n}$, $i = 1 \ldots 4$. So the maximum likelihood of seeing the counts (n_1, n_2, n_3, n_4) is

$$\hat{p}(n_1, n_2, n_3, n_4) = \frac{n!}{n_1! n_2! n_3! n_4!} \hat{p}_1^{n_1} \hat{p}_2^{n_2} \hat{p}_3^{n_3} \hat{p}_4^{n_4}.$$

Under the assumption that the relevance or irrelevance of a randomly selected document is independent of the presence or absence of a given term, the probabilities of p_1,p_2,p_3, and p_4 can be estimated by Eqs. (2)–(5) as shown in the previous section. The likelihood of seeing the counts under the independence assumptions is

$$\bar{p}(n_1, n_2, n_3, n_4) = \frac{n!}{n_1!n_2!n_3!n_4!} \bar{p}_1^{n_1} \bar{p}_2^{n_2} \bar{p}_3^{n_3} \bar{p}_4^{n_4}.$$

The log of the ratio of $\hat{p}$ over $\bar{p}$ is

$$\begin{aligned} w_t &= \log \frac{\hat{p}}{\bar{p}} = \log \frac{\frac{n!}{n_1!n_2!n_3!n_4!} \hat{p}_1^{n_1} \hat{p}_2^{n_2} \hat{p}_3^{n_3} \hat{p}_4^{n_4}}{\frac{n!}{n_1!n_2!n_3!n_4!} \bar{p}_1^{n_1} \bar{p}_2^{n_2} \bar{p}_3^{n_3} \bar{p}_4^{n_4}} \\ &= n_1 \log \left(\frac{\hat{p}_1}{\bar{p}_1} \right) + n_2 \log \left(\frac{\hat{p}_2}{\bar{p}_2} \right) + n_3 \log \left(\frac{\hat{p}_3}{\bar{p}_3} \right) + n_4 \log \left(\frac{\hat{p}_4}{\bar{p}_4} \right) \\ &= n_1 \log \frac{n_1}{e_1} + n_2 \log \frac{n_2}{e_2} + n_3 \log \frac{n_3}{e_3} + n_4 \log \frac{n_4}{e_4}. \end{aligned}$$

The expected counts e_1, e_2, e_3, and e_4 are given in the previous section. When this measure is used for term selection, the terms in the relevant documents are ranked by w_t in descending order, and the top-ranked terms are chosen for query expansion. The likelihood ratio is closely related to the maximum likelihood ratio test for multinomial distribution. We refer readers to Chapter 9 in Rice (1995) for the treatment of likelihood ratio test for the multinomial distribution, and Chi-square test.

3.1.5. Relative frequency ratio. The relative frequency ratio is the ratio of the relative frequency of a term in the relevant documents treated as one unified document over that in the whole collection. The relative frequency of a term in the collection is the number of times the term occurs in the collection over the number of terms in the collection. The rationale for using the relative frequency ratio to rank terms is that the terms that frequently occur in the relevant documents, but infrequently in the whole collection are assumed to be more useful in retrieving relevant documents than other terms. The log of relative frequency ratio for term t is computed as follows:

$$w_t = \log \frac{\frac{rtf_t}{rl}}{\frac{ctf_t}{cl}} \tag{6}$$

where ctf_t is the count of the occurrences of term t in the collection, cl is the collection length, i.e., the total number of occurrences of all the terms in the collection, rl is the number of occurrences of all the terms in the top-ranked documents that are assumed relevant, i.e., the sum of the document length over the set of relevant documents, and rtf_t is the total number of occurrences of term t in all the top-ranked documents that are assumed relevant. If we treat the collection as one document and all the relevant documents as another, then the weight assigned to term t is just the ratio of the relative frequency of term t in the combined relevant document over that of the term t in the whole collection.

For every term t, except for stopwords, found in the top-ranked documents that are presumed relevant, we compute its weight w_t according to one of the five term selection methods presented in the previous subsection. Then all the terms are ranked in decreasing order by their weight w_t. The top-ranked terms are added to the initial query to create a new query. Some of the selected terms may be among the initial query terms.

3.2. Query term weighting of selected terms

Once terms have been selected, we have the task of assigning them weights (query term weights) for the expanded query. For the selected terms that are not in the initial query, the weight is set to 0.5. The rationale for assigning weights to the selected terms that are not in the initial query is that the selected terms are considered not as important as the initial query terms, so the weights assigned to them should fall in the range of 0 to 1, exclusive. In our implementation, we set the weights of the new terms to 0.5, expecting that the query length would be doubled after query expansion. We separated the procedure for term selection and that for term weighting so that the current term selection procedure could be easily replaced with another one without changing the rest in query expansion. An alternative way of assigning weights to the new terms (the selected ones that are not in the initial query) is to use the term selection weights (e.g., the relevance weighting values, or Chi-square values) that were used to rank the terms in the term selection procedure. To use the weights computed in term selection, one would need to normalize the weights so that they fall in the range of 0 and 1. For those selected terms that are in the initial query, the weight is set to $0.5 * t_i$, where t_i is the occurrence frequency of term t in the initial query. The weights are unchanged for the initial query terms that are not in the set of selected terms. The selected terms are combined with the initial query terms to formulate an expanded query. When a selected term is one of the query terms in the initial query, its weight in the expanded query is the sum of its weight in the initial query and its weight assigned in the term selection process. For a selected term that is not in the initial query, its weight in the final query is the same as the weight assigned in the term selection process, which is 0.5. The weights for the initial query terms that are not among the selected terms remain unchanged. An example is presented below to illustrate how the expanded query is created from the initial query and the selected terms.

Initial query	Selected terms	Expanded query
t_1 (1.0)		t_1 (1.0)
t_2 (2.0)	t_2 ($2 * 0.5$)	t_2 (3.0)
t_3 (1.0)	t_3 ($1 * 0.5$)	t_3 (1.5)
	t_4 (0.5)	t_4 (0.5)

The numbers in parentheses are term weights. For example, the weight for term t_3 in the expanded query is 3.0, since it is in the initial query with a weight value of 2.0 and it is one of the selected terms assigned the weight of $2 * 0.5$.

Three minor changes are made to the blind relevance feedback procedure described in Section 3. First, a constant of 0.5 was added to every item in the formula used to compute the weight. Second, the selected terms must occur in at least 3 of the top-ranked documents that are presumed relevant. Third, the top-ranked two documents after the initial search remained as the top-ranked two documents in the final search. That is, the final search does not affect the ranking of the first two documents after the initial search. The rationale for not changing the top-ranked few documents is that when a query has only a few relevant documents in the entire collection and if they are not ranked in the top after the initial search, it is unlikely these few relevant documents would be risen to the top in the second search since most of the documents that are presumed relevant are actually irrelevant. On the other hand, if these few relevant documents are ranked in the top after the initial search, after expansion, they are likely to be ranked lower in the final search for the same reason. We believe a good strategy is to not change the ranking of the top few documents. In our implementation, we chose not to change the ranks of the top two documents in the final search.

Note that in computing the relevance probability of a document with respect to a query in the initial search, the ql is the number of terms in the initial query, and qtf_t is the number of times that term t occurs in the initial query. After query expansion, qtf_t is no longer the raw term frequency in the initial query, instead it is now the weight of term t in the expanded query, and ql is the sum of the weight values of all the terms in the expanded query. For the example presented above, qtf_{t_3} is 1.5, and ql is 6.0 (i.e., $1.0 + 3.0 + 1.5 + 0.5$). The relevance clues related to documents and the collection are the same in computing relevance probability using the expanded query as in computing relevance probability using the initial query. The number of selected terms is approximately twice the average number of unique terms in the original topics. Adding too many terms may decrease the importance of the original query terms since the relative frequencies of the original query terms decrease as the expanded query gets longer.

4. Decompounding

Compounds are words formed by joining two or more short words. For English, one way to create a compound word is to join directly two or more short words. Another way is to join two or more short words together with hyphens separating them, such as *second-guess*. Compounds occur in natural language texts, frequently in some languages such as German, but less so in others such as English. It is not difficult to find two-word compounds in English texts. Some examples are *breathtaking*, *birthday*, *blackmail*, and *whereabouts*. However, English compounds of three or more words are much less common. In English, the long compounds are formed by joining words together with hyphens separating the component words, such as *body-builder-turned-actor* and *in-between-age-children*. Some of the compound words, such as *birthday*, are compositional. The meaning of a compositional compound can be derived from the meanings of the component words that make up the compound. Some compound words, such as *blackmail* and *copycat*, are obscure, metaphorical, or non-compositional. The meaning of a non-compositional compound word cannot be derived from the meanings of its component words. In German texts, unlike in

English, compound words are common and compounding is a productive process. Most German compounds are formed by directly joining two or more words. Such examples are *Computerviren* (computer viruses), which is the concatenation of *Computer* (computer) and *Viren* (viruses). Sometimes an additional letter such as *s* is inserted between two words. For example, the compound *Schönheitskönigin* (beauty queen) is derived from *Schönheit* and *königin* with *s* inserted between them. There are also cases where compounds are formed with the final letter *e* of the first word elided. As an example, the compound *Erdbeben* (earthquake) is formed from *Erde* (earth) and *Beben* (trembling).

We present a German decompounding procedure in this section which will only address the cases where the compounds are directly formed by joining words and the case where the additional letter *s* is inserted between words. The procedure can be described as follows:

1. Create a German base dictionary consisting of German non-compound words in various forms.
2. Decompose a German compound with respect to the base dictionary. That is, find all possible ways to break up a compound with respect to the base dictionary.
3. Choose the decomposition with the smallest number of component words.
4. If there is more than one decomposition with the smallest number of component words, then choose the one with the highest probability of decomposition. The probability of a decomposition is estimated by the product of the relative frequencies of the component words. More details on computing decomposition probability are presented below.

For example, when the German base dictionary contains *ball, europa, fuss, fussball, meisterschaft* and others, the German compound *fussballeuropameisterschaft* (European Football Cup) can be decomposed into component words with respect to the base dictionary in two different ways as shown in the following table.

	Decompositions			
1	fuss	ball	europa	meisterschaft
2	fussball	europa	meisterschaft	

The second decomposition has the smallest number of component words, so the German compound *fussballeuropameisterschaft* is split into *fussball*, *europa* and *meisterschaft*. As another example, the following table shows the decompositions of the German compound *wintersports* (winter sports) with respect to a base dictionary containing *port, ports, s, sport, sports, winter, winters*, and others.

	Decompositions			$\log p(D)$
1	winter	s	ports	−43.7002
2	winter	s	ports	−20.0786
3	winters	s	ports	−28.3584

The compound *wintersports* has three decompositions with respect to the base dictionary. Because all three decompositions have the same number of component words (the letter *s* appearing between component words is not considered as a component word), the rule of selecting the decomposition with the smallest number of component words cannot be applied here. We have to compute the probability of decomposition for all three decompositions. The last column in the above table shows the log of the decomposition probabilities of all three decompositions that were computed using relative frequencies of the component words in the German test collection of CLEF 2002. According to the rule of selecting the decomposition of the highest probability, the second decomposition should be chosen as the decomposition of the compound *wintersports*. That is, the compound *wintersports* should be split into *winter* and *sports*.

To compute the probability of a decomposition, consider the decomposition of a compound c into n component words, $c = w_1 w_2 \ldots w_n$. The probability of a decomposition is computed as follows:

$$p(c) = p(w_1)p(w_2)\ldots p(w_n) = \prod_{i=1}^{n} p(w_i)$$

where the probability of component word w is computed as follows:

$$p(w_i) = \frac{ctf_{w_i}}{\sum_{j=1}^{N} ctf_{w_j}}$$

where ctf_{w_i} is the number of occurrences of word w_i in a collection, N is the number of unique words, including compounds, in the collection. The occurrence frequency of a word is the number of times the word occurs alone in the collection. The frequency count of a word does not include the cases where the word is a component word in a longer compound. Also, the base dictionary does not contain three-letter or shorter words except the letter *s*. We created a German base dictionary by combining a lexicon extracted from Morphy, a German morphological analyzer (Lezius et al. 1998), German wordlists found on the Internet, and short German words in the CLEF 2001 German collection. Morph decomposes only compound nouns using longest-matching rules (Lezius et al. 1998). In our implementation, we considered only the case where a compound is the concatenation of component words, and the case where the letter *s* is inserted between component words.

In general, breaking up compounds is helpful. The same phrase may be spelled out in words sometimes, but as one compound other times. When a user formulates a German query, the user may not know if a phrase should appear as multi-word phrase or as one compound. An example is the German equivalent of the English phrase "European Football Cup", in the *title* field of topic 113, the German equivalent is spelled as one compound *Fussballeuropameisterschaft*, but in the *description* field, it is *Europameisterschaft im Fußball*, yet in the *narrative* field, it is *Fußballeuropameisterschaft*. This example brings out two points in indexing German texts. First, it should be helpful to split compounds into component words. Second, normalizing the spelling of ss and ß should be helpful. The German equivalent for "Nobel prize winner for literature" is *Literaturnobelpreisträger*, but

in the "Der Spiegel" German collection, we find variants of *Literatur-Nobelpreisträger* and *Literaturnobelpreis-Trägerin*. One more reason why decompounding is desirable is that when an English phrase is translated into German, the German translation may be a compound, but it could also be a multi-word phrase. For example, when the English phrase "Latin America" was translated into German using Babelfish, its German translation was *lateinischem Amerika*. However, the more common form is the compound *Lateinamerika*. In translating German into English, one may see cases where the German compounds cannot be translated, yet the component words can be translated separately. For example, the German compound *Bronchialasthma* (bronchial asthma) was not properly translated into English, however the component words were.

It is not always desirable to split up German compounds into their component words. Consider the compound *Erdbeben*. In this case, it is probably better not to split up the compound. But in other cases like the compound *Gemüseexporteure* (vegetable exporters) in topic 90 and the compound *Fußballweltmeisterschaft* (World Soccer Championship) in topic 51, splitting up the compounds probably is beneficial since the use of the component words might retrieve additional relevant documents which are otherwise likely to be missed if only the compounds are used. In fact, we noticed that the compound *Gemüseexporteure* does not occur in the CLEF 2001 German document collection.

Monz and de Rijke (2002) present a procedure for splitting German noun-noun compounds. It splits a compound by recursively removing the prefix that is a noun found in the lexicon from the remaining of the compound if the substring after removing the prefix can be decomposed into words in the lexicon. Savoy (2002b) proposes a German decompounding procedure based on a set of pre-defined patterns.

5. Fast document translation

To translate a large collection of documents from a source language to a target language using a machine translation system can be computationally intensive and may take a long time. In this section we present an approximate but fast approach to translating source documents into a target language. We first collect all the unique words in the source documents, then translate the source words individually into the target language using a machine translation system. Once we have the translations of all the source words, we can translate a source document into the target language by replacing the words in the source document with their translations in the target language. The translation is only approximate, but very fast. It is approximate since the same source word is always translated into the same target word. Obviously when a source word has multiple meanings under different contexts in the source documents, the translations of the source word may not be the same in the target language. For example, in translating English into French, the English word *race* is translated into the French word *race*. However, the English word *race* is polysemous, it could mean *human race* or *race in sports*. When it means *race* in sports, the appropriate French translation is not *race*, but *course*. For multilingual retrieval, one can translate the document collections into the topic language using this method if one can find a MT system capable of translating documents into the topic language. For example, to perform searching English topics against a collection of documents in English, French, German, Italian, and

Spanish, one can translate the French, German, Italian, and Spanish documents into English using the bilingual wordlists derived from the source documents and a MT system. When the documents are translated into English, one can index the English documents and the translated English documents from other languages together. One benefit of translating documents word-by-word using bilingual wordlists created from MT systems is that the translation can be very fast. Bilingual wordlists created from parallel texts or from bilingual dictionaries can also be used to translate documents by translating individually document words. In this case, there is no need to merge documents rankings from different languages. McCarley and Roukos (1998) present a different approach to fast translating documents into query language. Their system is based on a statistical machine translation model and trained on a parallel corpus. Our approach uses commercial machine translation systems to translate document words into the query language first, then translate documents word-by-word into the query language.

6. Test collections

The document collection for the multilingual IR task at both CLEF 2001 and CLEF 2002 consists of documents in five languages: English, French, German, Italian, and Spanish. The collection has about 750,000 documents which are newspaper articles published in 1994 except that part of the *Der Spiegel* was published in 1995. A set of 100 topics, 50 for CLEF 2001 and 50 for CLEF 2002, was developed and released in more than 10 languages, including Dutch, English, French, German, Italian, and Spanish. The topics were numbered 41 through 90 for CLEF 2001, and 91 through 140 for CLEF 2002. A topic has three parts: (1) *title*, a short description of information need; (2) *description*, a sentence-long description of information need; and (3) *narrative*, a more lengthy description specifying document relevance criteria. We will refer to the document collection and the topics numbered from 41 to 90 as the *CLEF 2001 test collection*, and the same document collection and the topics numbered from 91 to 140 as the *CLEF 2002 test collection*. For more details about the test collections, see (Braschler and Peters, this volume). The multilingual IR task at both CLEF 2001 and CLEF 2002 is concerned with searching the collection consisting of English, French, German, Italian, and Spanish documents for relevant documents, and returning a combined, ranked list of documents in any document language in response to a query. All retrieval runs reported below used only the *title* and *description* fields in the topics.

7. Indexing

Indexing of the texts consists of seven steps: *pre-processing*, *tokenization*, *normalization*, *stopwords removal*, *decompounding*, *stemming*, and *post normalization*. The seven steps are sequentially carried out in the order as presented above. Not all seven steps are executed in indexing for every language. Some of the steps are optional. For example, *pre-processing* is only applied to Italian texts to restore the accents from the source Italian texts, and decompounding is applied only to German to split German compounds into their component words. The indexing procedure is designed to work directly on the source documents. Both topics and documents are indexed in the same way.

The source texts are broken into tokens in the *tokenization* process. A token can contain only valid characters which include the digits and letters in the ISO 8859-1 (Latin-1) characters set. Characters that are not in the valid characters set are treated as word delimiters. The *normalization* step changes upper-case letters into lower case, including the upper-case letters with diacritic marks. Stopwords in both documents and topics are removed from indexing. We have two stoplists for each of the five languages, one for indexing documents and the other for indexing topics. The stoplist for topics contains all the stopwords for documents and some additional stopwords such as *relevant* and *document*. For German, compounds are replaced by their component words in both documents and topics. Only the component words are retained. The Muscat multiple language stemmer is applied to the remaining words. The Muscat stemmer set includes stemmers for English, French, German, Italian, Spanish, and others. The Muscat stemmer sets are rule-based stemmers which have lately evolved into the SNOWBALL stemmer generation language developed by Martin Porter (2001). They may be obtained at http://snowball.tartarus.org/. The last step in indexing removes the diacritic marks. The diacritic marks are not removed in the *normalization* step because the stemmers may utilize the diacritic marks.

8. Experimental results

8.1. Evaluation of term selection methods for query expansion

To test the effectiveness of the five term selection methods described in Section 3.1, we performed a series of monolingual retrieval runs using CLEF 2001 and CLEF 2002 test collections for five languages. Table 1 presents the evaluation results of the five term selection methods. The results shown in the last six columns in the table are the average precision values for all the runs. Only the *title* and *description* fields were used in all the runs. The

Table 1. Evaluation of blind feedback, term selection methods.

Language	Test collection	Topic fields	No. topics	Baseline	Term selection methods				
					RW	RFR	MI	CHI	LRM
English	CLEF 2001	T, D	47	0.5229	0.5474	**0.5557**	0.5472	0.5455	0.5489
French	CLEF 2001	T, D	49	0.4713	0.5082	**0.5171**	0.5156	0.5137	0.5063
German	CLEF 2001	T, D	49	0.4329	**0.4825**	0.4779	0.4751	0.4772	0.4799
Italian	CLEF 2001	T, D	47	0.4510	0.4881	0.4871	0.4913	0.4933	**0.4958**
Spanish	CLEF 2001	T, D	49	0.5327	0.5712	0.5714	0.5722	**0.5756**	0.5747
English	CLEF 2002	T, D	42	0.5084	0.5602	0.5642	0.5621	0.5572	**0.5698**
French	CLEF 2002	T, D	50	0.4347	**0.5191**	0.5128	0.5133	0.5151	0.5137
German	CLEF 2002	T, D	50	0.4393	0.5234	0.5256	0.5230	**0.5280**	0.5186
Italian	CLEF 2002	T, D	49	0.4169	0.4750	**0.4773**	0.4734	0.4698	0.4635
Spanish	CLEF 2002	T, D	50	0.5016	0.5338	0.5334	0.5290	0.5312	**0.5341**

column labeled *no. topics* gives the number of topics having at least one relevant document in the test collection. The topics with no relevant documents were excluded in computing average precision. The column labeled *baseline* presents the average precision of a monolingual run without query expansion. When query expansion was performed, the top-ranked 10 documents after the initial search using the initial query were assumed relevant. Then all the terms, excluding stopwords, in the presumed relevant documents were weighted using one of the five term selection methods and then ranked in descending order by their selection weights. The top-ranked 10 terms were selected for query expansion. The procedure for assigning query weights to the selected terms and combining the selected terms with the initial query terms was presented in Section 3.2. The *language* and *test collection* columns in the table together indicate which document collection and which topic set were used in a monolingual run. As an example, the pair *English* and *CLEF 2001* means that the 50 English topics and the English documents in CLEF 2001 test collection were used in the monolingual retrieval. Each row presents the average precision values for six monolingual runs using the same set of topics and the same document collection, one for the monolingual run without query expansion and five runs (one for each of the five term selection methods) with query expansion. The best precision among the runs with query expansion is presented in bold face for the same topics and documents. The results presented in Table 1 show that the performances of all five term selection methods used in query expansion are very close to each other for the same set of topics and documents. Furthermore, there is no single winner among the five term selection methods. No single method is consistently superior than the others. Since 10 terms are selected from the top-ranked 10 documents after the initial search, a total of 1,000 terms are selected for the 100 topics in each language. The number of topic-terms selected by all five term selection methods is 510 (51.0%) for English, 530 (53.0%) for German, 554 (55.4%) for French, 574 (57.4%) for Italian, and 536 (53.6%) for Spanish. Overall, slightly over half of the terms selected by any one of the five term selection methods are also selected by the other four term selection methods. A term is considered selected by two methods if it is selected for the same topic by both methods in query expansion. For all retrieval runs with query expansion in the remainder of this paper, the *relevance weighting* (RW) method was used to select terms for query expansion.

8.2. Evaluation of monolingual retrieval

8.2.1. Query expansion. In this section we present the results of monolingual retrieval. For automatic query expansion, the top-ranked 10 terms from the top-ranked 10 documents after the initial search were combined with the original query to create the expanded query. For German monolingual runs, the compounds were split into their component words using the procedure described in Section 4, and only the component words were retained in both document and topic indexes. All the monolingual runs included automatic query expansion via the relevance feedback procedure described in Section 3. Table 2 gives the monolingual retrieval results for five document languages. The last column labeled *change* shows the improvement of average precision with blind relevance feedback over without it. As Table 2 shows, query expansion increased the average precision of the monolingual runs for all five languages, the improvement ranging from 6.42% for Spanish to 19.42% for French using

Table 2. Monolingual retrieval performances on CLEF 2001 and CLEF 2002 test collections.

Run id	Test collection	Language	No. topics	Topic fields	Without expansion		With expansion		Change (%)
					Overall recall	Average precision	Overall recall	Average precision	
moen1	CLEF 2001	English	47	T, D	820/856	0.5229	839/856	0.5474	4.69
mofr1	CLEF 2001	French	49	T, D	1189/1212	0.4713	1201/1212	0.5082	7.83
mode1	CLEF 2001	German	49	T, D	2009/2130	0.4329	2038/2130	0.4825	11.46
moit1	CLEF 2001	Italian	47	T, D	1200/1246	0.4510	1219/1246	0.4881	8.23
moes1	CLEF 2001	Spanish	49	T, D	2549/2694	0.5327	2596/2694	0.5712	7.23
moen2	CLEF 2002	English	42	T, D	765/821	0.5084	793/821	0.5602	10.19
mofr2	CLEF 2002	French	50	T, D	1277/1383	0.4347	1354/1383	0.5191	19.42
mode2	CLEF 2002	German	50	T, D	1696/1938	0.4393	1807/1938	0.5234	19.14
moit2	CLEF 2002	Italian	49	T, D	994/1072	0.4169	1024/1072	0.4750	13.94
moes2	CLEF 2002	Spanish	50	T, D	2531/2854	0.5016	2673/2854	0.5338	6.42

the CLEF 2002 topics, and from 4.69% for English to 11.46% for German using the CLEF 2001 topics. The topics having no relevant documents in the test collection were excluded in computing the average precision. The number of topics having at least one relevant document is presented in the column labeled *no. topics*. In our implementation of blind relevance feedback, the top-ranked two documents after the initial search remain as the top-ranked two documents in the final search. Without this constraint, the average precision for all five languages changed very little. Using the CLEF 2002 test collection, the average precision was 0.5363 for English, 0.5143 for French, 0.5170 for German, 0.4842 for Italian, and 0.5408 for Spanish when the constraint was not imposed. There are 37 topics for all five languages with only one to five relevant documents in the CLEF 2002 test collection. The average precision for those 37 topics was 0.5386 without query expansion, 0.5629 with query expansion and the constraint, and 0.5137 with query expansion but without the constraint. For query expansion, the top-ranked 10 terms were selected from the top-ranked 10 documents after the initial search, which implies that at least half of the top-ranked 10 documents that were assumed relevant are, in fact, not relevant since each topic in this set of 37 topics has only one to five relevant documents in the collection. Not changing the ranks of the top-ranked two documents after query expansion increased the average precision for those topics with one to five relevant documents. However, when all topics are considered, leaving the top two document ranks unchanged has little effect on average precision over all topics, including the topics with more than five relevant documents.

The documents in CLEF 2001 test collection and CLEF 2002 test collection are the same, only the topics are different. When we conducted the t test on the hypothesis that query expansion has no impact on retrieval performance, we used the topics in both CLEF 2001 test collection and CLEF 2002 test collection, instead of carrying out one t test for the topics in CLEF 2001 and another one for the topics in CLEF 2002. The p-value of the t test for English topics is 0.0021, and the p-values for the other four languages are much

smaller than 0.01. So the performance difference between retrieval with query expansion and without it is significant.

The significance testing we carry out in this paper is the paired t-test and the t-statistic is computed as presented by Hull (1993). The null hypothesis is that two methods or techniques being tested are equally effective in terms of retrieval performance measured by recall and precision. When the p-value of a paired t-test is below 0.05, we conclude that the difference in retrieval effectiveness between the two methods is significant.

The following table gives the number of topics with positive, negative, or no effects by query expansion.

Language	No. topics	$AP_{\text{with}} > AP_{\text{without}}$	$AP_{\text{with}} = AP_{\text{without}}$	$AP_{\text{with}} < AP_{\text{without}}$
English	89	47	12	30
French	99	63	9	27
German	99	68	4	27
Italian	96	66	1	29
Spanish	99	62	3	34

Column 3 shows the number of topics for which query expansion increased performance, column 4 shows the number of topics for which query expansion had no impact, and the last column shows the number of topics for which query expansion degraded the performance. The column labeled *no. topics* presents the number of topics having at least one relevant document in CLEF 2001 or CLEF 2002 test collection. AP_{with} denotes the average precision of a run with query expansion, and AP_{without} denotes the average precision of a run without query expansion.

8.2.2. Effects of diacritic marks. We performed a series of monolingual retrieval runs to test the impact of removing the diacritic marks on retrieval effectiveness. Table 3 presents the evaluation results of 10 monolingual runs using CLEF 2002 test collection. Two retrieval runs were carried out for each language, one without accent normalization and one with

Table 3. Effect of accent normalization on monolingual retrieval performances.

				Without accent normalization		With accent normalization		
Test collection	Language	No. topics	Topic fields	Overall recall	Average precision	Overall recall	Average precision	Change (%)
CLEF 2002	English	42	T, D	741/821	0.4719	741/821	0.4838	2.52
CLEF 2002	French	50	T, D	1195/1383	0.3622	1203/1383	0.3847	6.21
CLEF 2002	German	50	T, D	1346/1938	0.3392	1359/1938	0.3462	2.06
CLEF 2002	Italian	49	T, D	957/1072	0.3733	961/1072	0.3817	2.25
CLEF 2002	Spanish	50	T, D	2456/2854	0.4442	2490/2854	0.4611	3.80

accent normalization. For these 10 monolingual runs, words were changed to lower case and stopwords were removed, but neither stemming nor query expansion was applied. The last column labeled *change* shows the change of average precision with accent normalization over without it. With accent normalization, for all five languages, the average precision was increased, ranging from 2.06% for German to 6.21% for French. The precision was increased over 0.1 after removing diacritic marks for only 9 out of 241 topics for all five languages with at least one relevant document. For the majority of the topics regardless of the language, removing diacritic marks had little impact on retrieval effectiveness, however, for a small number of topics, the increase of average precision due to the removal of diacritic marks was substantial. As an example, the precision for Spanish topic 114 with the title "Guerra Civil en Afganistán" was increased from 0.0583 without removing diacritic marks to 0.1442 with removing diacritic marks, and the number of relevant documents retrieved was increased from 20 to 39 out of a total of 45 relevant documents. Another example is Spanish topic 98 with the title "Películas de los Kaurismäki", after removing diacritic marks, the precision was increased from 0.4008 to 1.0. This topic has 5 relevant documents which were all retrieved and ranked in the top when diacritic marks were removed. For the few topics with substantial increase of precision after removing diacritic marks, the main reason is that the topic words with diacritic marks, such as *Kaurismäki*, sometimes occur in the documents without the diacritic marks. Since it is not common to see diacritic marks in English texts, we expect that removing diacritic marks would have very little effect on retrieval performance. Out of the 42 English topics with at least one relevant document in the CLEF 2002 test collection, removing diacritic marks affected the precision for only one topic, English topic 98, whose precision was increased from 0.5 to 1.0. The precision values for the other 41 English topics were the same before and after the removal of diacritic marks. We performed a t test for each language, comparing the performance with accent normalization over without it. Only the p-value of 0.0389 for French is below 0.05, the p-values of the t tests for English, German, Italian, and Spanish are 0.3232, 0.1295, 0.1699, and 0.1688, respectively, which are all above 0.05. We conclude that removing diacritic marks improved the retrieval performance, but not significantly except for French.

8.3. Evaluation of decompounding

For the German monolingual runs, compounds were decomposed into their component words by applying the decompounding procedure described in Section 4. Only component words of the decomposed compounds were kept in document and topic indexes. One of the 50 German topics in the CLEF 2001 test collection has no relevant German documents. The average precision values presented in Table 4 were computed with that topic excluded. The total number of German relevant documents for the remaining 49 topics for CLEF 2001 is 2130. Table 4 presents the results of German monolingual retrieval on CLEF 2001 test collection under different combinations of three features: *decompounding*, *stemming*, and *query expansion*. The features are implemented in the order of *decompounding*, *stemming*, and *query expansion*. For example, when *decompounding* and *stemming* are present, the compounds are split into component words first, then the components are stemmed. Stopwords were removed for all runs. When only stopwords were removed, the average precision

Table 4. German monolingual retrieval performance on CLEF 2001 test collection.

Features	None (1)	Decomp (2)	Stem (3)	Expan (4)	Decomp + stem (5)	Decomp + expan (6)	Stem + expan (7)	Decomp + stem + expan (8)
Avg prec	0.3380	0.3953	0.3842	0.4107	0.4329	0.4363	0.4581	0.4825
Change	baseline	+16.95%	+13.67%	+21.51%	+28.08%	+29.08%	+35.53%	+42.75%
Recall	1803	1910	1920	1869	2009	1943	2012	2038

Table 5. German monolingual retrieval performance on CLEF 2002 test collection.

Features	None (1)	Decomp (2)	Stem (3)	Expan (4)	Decomp + stem (5)	Decomp + expan (6)	Stem + expan (7)	Decomp + stem + expan (8)
Avg prec	0.3462	0.3859	0.3633	0.4145	0.4393	0.4517	0.4393	0.5234
Change	baseline	+11.47%	+4.94%	+19.73%	+26.89%	+30.47%	+26.89%	+51.18%
Recall	1359	1577	1500	1575	1696	1752	1702	1807

is 0.3380, which is considered as the baseline performance for the purpose of comparison. When any one of the three features was present, the performance increase ranged from 13.67% to 21.51%. When two of the three features were present, the performance increase ranged from 28.08% to 35.53%. The average precision was increased by 42.75% when all three features were present. Table 5 gives the performance of German monolingual retrieval on the CLEF 2002 German test collection under different combinations of three features. The stopwords were removed first for all the runs presented in the table. The total number of German relevant documents for 50 topics in CLEF 2002 test collection is 1938. The baseline performance obtained when only stopwords were removed was 0.3462. The table shows when any one of the three features is present, the average precision improves from 4.94% to 19.73% over the baseline performance when none of the features is present. When two of the three features are included in retrieval, the improvement in precision ranges from 26.89% to 30.47%. And when all three features are present, the average precision is 51.18% better than the baseline performance. It is interesting to see that the three features are complementary. That is, the improvement contributed by each individual feature is not diminished by the presence of the other two features. Without decompounding, stemming alone improved the average precision by 4.94%. However with decompounding, stemming improved the average precision from 0.3859 to 0.4393, an increase of 13.84%. Stemming became more effective because of decompounding. Decompounding alone improved the average precision by 11.47% for German monolingual retrieval using the CLEF 2002 topics, and by 16.95% using the CLEF 2001 topics. The t test results show that the performance with decompounding was significantly better than that without decompounding.

Table 6 shows some of the German words in the *title* or *desc* fields of the CLEF 2002 topics that were split into component words using the decompounding procedure described in Section 4. The column labeled *component words* shows the component words of the

Table 6. Some of the German words in *title* or *desc* fields of the topics that are split into component words.

Compounds	Component words		
1 computeranimationen	computer	animationen	
2 eurofighter	euro	fighter	
3 interessenkonflikts	interessen	konflikts	
4 fussballeuropameisterschaft	fussball	europa	meisterschaft
5 literaturnobelpreisträgers	literatur	nobel	preisträgers
6 schönheitswettbewerben	schönheit	s	wettbewerben

decomposed compounds. As an example, the compound *computeranimationen* (computer animation) was split into component words *computer* and *animationen*. The German word *eurofighter* was split into *euro* and *fighter* since both component words are in the base dictionary, but not the word *eurofighter*. Including the word *eurofighter* in the base dictionary will prevent it from being split into component words. Two topic words, *lateinamerika* (Latin America) and *zivilbevölkerung* (civil population), were not split into component words because both are present in our base dictionary which is far from perfect. For the same reason, the *preisträgers* (prize winner) was not decomposed into *preis* and *trägers*. An ideal base dictionary should contain all and only the words that should not be further split into smaller component words. Our current decompounding procedure does not split the words in the base dictionary into smaller component words. The topic word *südjemen* (southern Yemen) was not split into *süd* and *jemen* because our base dictionary does not contain words that are three-letter long or shorter. The majority of the errors in decompounding are caused by the incompleteness of the base dictionary or the presence of compound words in the base dictionary.

With stemming and query expansion, decompounding increased the precision for 60 topics, decreased the precision for 36 topics, and had no effect on 3 topics. The average increase in precision for the 99 topics was 0.0716. For 12 topics, the increase in precision was over 0.3, but for only one topic, the decrease in precision was over 0.3. For example, the precision for topic 109 entitled "Computersicherheit" was increased from 0.0006 without decompounding to 0.5166 with decompouding, and the precision for topic 105 entitled "Bronchialasthma" was increased from 0.1884 to 0.6399 after decompounding. The poor performance of topic 109 without decompounding can be attributed to the fact that the compound *Computersicherheit*, the most important term in topic 109, does not occur in the German collection while the component words, *computer* and *sicherheit*, occur 4,270 times and 8,513 times, respectively, in the German collection. The precision for topic 79 with the title "Raumsonde Odysseus" was decreased from 0.8486 to 0.2310 after decompouding. The word *Raumsonde* was split into *raum* and *sonde* after decompounding. The component word *sonde* became *sond* after stemming. While the stem, *raumsond*, of the compound *Raumsonde* occurs only 49 times in the German collection, the stems of the component words, *raum* and *sond*, occur 36,793 times and 51,899 times, respectively, in the collection. Without stemming, the precision might be better since the component word *sonde* occurs only 178 times in the collection. Topic 79 has 12 relevant documents in the collection.

8.4. Bilingual retrieval using MT

A major factor affecting the performance of bilingual retrieval and multilingual retrieval is the quality of translation resources. Two of the issues in dictionary-based CLIR are (1) determining the number of translations to retain when multiple candidate translations are available; and (2) assigning weights to the selected translations (Grefenstette 1998). When machine translation systems are used to translate topics, these two issues are resolved automatically by machine translation systems, since they provide only one translation for each word. However, when bilingual dictionaries or parallel corpora are used to translate topics, often for a source word, there may be several alternative translations.

In this section, we evaluate two machine translation systems, *online Babelfish translation* available at http://babelfish.altavista.com/ and *L&H Power Translator Pro, version 7.0*, for translating topics in bilingual retrieval. We used both machine translation systems to translate the 50 English topics for CLEF 2001 and the 50 English topics for CLEF 2002 into French, German, Italian, and Spanish. For each language, both sets of translations were preprocessed in the same way. Table 7 presents the bilingual retrieval performances using the 50 English topics for CLEF 2002. Only the title and description fields in the topics were indexed. The last column in Table 7 shows the improvement of average precision with query expansion over without it. When both L&H Translator and Babelfish were used in bilingual retrieval from English to French, German, Italian, and Spanish, the translations from L&H Translator and the translations from Babelfish were concatenated by topic. The term frequencies in the combined topics were reduced by half so that the combined topics were comparable in length to the source English topics. Then the combined translations were used to search the document collection for relevant documents as in monolingual retrieval. For example, for the English-to-Italian run *bienit1*, we first translated the source English topics into Italian using L&H Translator and Babelfish. The Italian translations produced by

Table 7. Performances of bilingual retrieval runs on CLEF 2002 test collection.

Run id	Topic	Document	MT	Without expansion precision	With expansion precision	Change (%)
bienfr1	English	French	Babelfish + L&H	0.4118	0.4773	+15.91
bienfr2	English	French	Babelfish	0.3731	0.4583	+22.84
bienfr3	English	French	L&H	0.3951	0.4652	+17.74
biende1	English	German	Babelfish + L&H	0.3561	0.4479	+25.78
biende2	English	German	Babelfish	0.3229	0.4091	+26.70
biende3	English	German	L&H	0.3555	0.4449	+25.15
bienit1	English	Italian	Babelfish + L&H	0.3608	0.4090	+13.36
bienit2	English	Italian	Babelfish	0.3239	0.3634	+12.20
bienit3	English	Italian	L&H	0.3412	0.3974	+16.47
bienes1	English	Spanish	Babelfish + L&H	0.4090	0.4567	+11.66
bienes2	English	Spanish	Babelfish	0.3649	0.4108	+12.58
bienes3	English	Spanish	L&H	0.4111	0.4557	+10.85

Table 8. Effectiveness of decompounding in bilingual retrieval to German.

Topic language	Document language	MT system	Without decompounding average precision	With decompounding average precision	Change (%)
English	German	L&H Translator	0.2776	0.3009	8.4
English	German	Babelfish	0.2554	0.2906	13.78
French	German	Babelfish	0.2774	0.3092	11.46

L&H Translator and the Italian translations produced by Babelfish were combined by topic. Then the combined, translated Italian topics with term frequencies reduced by half were used to search the Italian document collection. The *bienfr1*, *biende1*, and *bienes1* bilingual runs from English were all produced in the same way as the *bienit1* run. For English to German bilingual retrieval runs, the words in *title* or *desc* fields of the translated German topics were decompounded. For all bilingual runs, words were stemmed after removing stopwords.

All the bilingual runs applied blind relevance feedback. The top-ranked 10 terms from the top-ranked 10 documents after the initial search were combined with the initial query to formulate an expanded query. The results presented in Table 7 show that query expansion improved the average precision from 10.85% to 26.70%. The L&H Translator performed better than Babelfish for bilingual retrieval from English to French, German, Italian, and Spanish. Combining the translations from L&H Translator and Babalfish performed slightly better than using only the translations from L&H translator.

We noticed a number of errors in translating English to German using Babelfish. For example, the English text *Super G* was translated into *Superg*; *U.S.-Russian* was not translated. While the phrase *Southern Yemen* in the *desc* field was incorrectly translated into *Südyemen*, the same phrase in the *title* field became *SüdcYemen* for some unknown reason. The correct translation should be *Südjemen*. Decompounding is helpful in monolingual retrieval, it is also beneficial in bilingual retrieval to German from other languages such as English. Table 8 shows the performances of three bilingual retrieval runs from English or French to German with and without decompounding. All three runs were performed without stemming or query expansion. The improvement because of decompounding in average precision ranges from 8.4% to 13.78%. One reason why indexing the component words of compounds instead of compounds is beneficial is that a multi-word English phrase may be translated into a multi-word German phrase, or into a compound. For example, in topic 109, the English phrase *Computer Security* became *Computer-Sicherheit* in the *title*, but the same phrase in lower case in the *desc* field became *Computersicherheit*.

8.5. Multilingual retrieval experiments

A common approach to multilingual retrieval is to first translate the source topics into all document languages, then carry out one monolingual retrieval for each language using the translated topics, and last combine the ranked lists of documents resulted from the monolingual runs into a unified ranked list of documents in all document languages.

The problem of merging multiple runs is closely related to the problem of calibrating the estimated probability of document relevance and the problem of estimating the number of relevant documents with respect to a given query in a collection. If the estimated probability of document relevance is well calibrated, that is, the estimated probability is close to the true probability of relevance, then it would be trivial to combine multiple runs into one, since all one needs to do will be to combine the multiple runs and re-rank the documents by estimated probability of relevance. If the number of relevant documents with respect to a given query could be well-estimated, then one could take the number of documents from each individual run that is proportional to the number of estimated relevant documents in each collection. Neither of these problems is easy to solve.

A fundamental difference between merging in monolingual retrieval or bilingual retrieval and merging in multilingual retrieval is that in monolingual or bilingual retrieval, documents for individual ranked lists are from the same collection, while in multilingual retrieval, the documents for individual ranked lists come from different collections. For monolingual or bilingual retrieval, if we assume that documents appearing on more than one ranked list are more likely to be relevant than the ones appearing on a single ranked list, then we should rank the documents appearing on multiple ranked lists in higher position in the merged ranked list of documents. A simple way to accomplish this is to sum the relevance values for the documents appearing on multiple ranked lists while the relevance values for the documents appearing on a single list remain the same. After summing up the relevance values, the documents are re-ranked in descending order by combined relevance values. In multilingual retrieval merging, since the documents on the individual ranked lists are all different, we cannot use multiple appearances of a document in the ranked lists as evidence to promote its rank in the final ranked list.

This section describes the multilingual retrieval experiments using the English topics (only *title* and *description* fields were indexed). As mentioned in the bilingual experiments section above, the 100 English topics for CLEF 2001 and CLEF 2002 were translated into the other four document languages: French, German, Italian, and Spanish, using both Babelfish and L&H Translator. We will compare three simple approaches to merging ranked lists of documents resulted from different monolingual runs. We will also compare four approaches to multilingual retrieval. In Section 8.5.3 we present the results of combining two multilingual retrieval runs, one based on query translations and one based on document translations. In Section 8.5.4 we present a procedure for computing optimal performance that could possibly be achieved by any merging algorithm under the constraint that the relative ranking order of the documents on all individual ranked lists is preserved in the final ranked list of documents.

8.5.1. Comparing merging methods. There are some simple ways to merge ranked lists of documents from different collections. The first approach, called *round-robin merging*, ignores the raw relevance values of the documents and considers solely the ranks of the documents in the individual ranked lists. To create the final ranked list, starting from the top, one takes one document from each ranked list of documents in a round-robin fashion and adds the documents to the final ranked list. An alternative approach is to first combine all the individual ranked lists, then sort the combined list by topic and rank. This approach is

also called *rank-based merging*. The second approach combines the individual ranked lists of documents, and then sorts the combined ranked list by the un-normalized raw relevance values. Thus, this approach is called *raw-score merging*. The third approach normalizes the relevance values of the documents before merging. There are a few techniques to normalize the relevance values. One technique divides the relevance value of a document with respect to a query by the maximum relevance value (the value of the first-ranked document for a topic) of the same topic. The second technique used in Savoy (2002a) normalizes relevance values as follows.

$$newrv_i = \frac{rv_i - rv_{\min}}{rv_{\max} - rv_{\min}}$$

where $rv_{\min}$ is the minimum relevance value for a topic, and $rv_{\max}$ the maximum relevance value for the same topic. The third technique also experimented in Savoy (2002a, 2002b) multiplies relevance values by a score computed for each individual collection. This approach is called *CORI* developed by Callan et al. (1995) for merging lists of documents from distributed collections. Recently Savoy (2002b) proposed a method for normalizing relevance values by considering both the relevance values and the ranks of documents. The new relevance value of a document is computed using its original relevance value and the log of its rank. The coefficients for the raw relevance value and the log of the rank of a document are determined by fitting a logistic regression model to training data. Many CLEF participanting groups used score-based or rank-based merging strategies. See for example Braschler et al. (2002), Chen (2002a), Kraaij (2002) and McNamee and Mayfield (2002).

A limitation of the *round-robin* merging is that it is prone to the skewed distribution of relevant documents over the document languages. Since, for each topic, the same number of documents for every document language is taken from the individual ranked lists of documents to create the final ranked list of documents, when the relevant documents for a topic are not evenly distributed across all document languages, it is highly likely that too many documents are taken from the collection with few relevant documents, and too few documents taken from the collection with many relevant documents. As an extreme example, if all the relevant documents for a topic are concentrated in one document language, the *round-robin* merging approach will still take the same number of documents for every document language to create the final ranked list. If the collection represents five document languages, then at least four out of five documents in the final ranked list are irrelevant. One weakness of the *raw-score* merging is that it is prone to incomparable relevance values.

Here we will evaluate three merging strategies. The first method is to combine all ranked lists, sort the combined list by the raw relevance score, then take the top 1000 documents per topic. The second method is to normalize the relevance score for each topic, dividing the relevance scores of the retrieved documents for a topic by the relevance score of the highest-ranked document for the same topic. Table 9 presents the multilingual retrieval performances with different merging strategies on the CLEF 2002 test collection. The multilingual runs were produced by merging five runs: moen2 (English-English, 0.5602), bienfr1 (English-French, 0.4773), biende1 (English-German, 0.4479), bienit1 (English-Italian, 0.4090), and bienes1 (English-Spanish, 0.4567). The bilingual runs, *bienfr1*, *biende1*, *bienit1* and *bienes1*, were described in Section 8.4; and the monolingual run *moen2* in Section 8.2. The

Table 9. Multilingual retrieval performances for different merging strategies on the CLEF 2002 test collection.

Run id	Topic language	Topic fields	Merging strategy	Recall	Precision
muena	English	T, D	Round-robin	5792/8068	0.3326
muenb	English	T, D	Raw-score	5880/8068	0.3762
muenc	English	T, D	Normalized-score	5765/8068	0.3570

topics having no relevant documents were not removed before merging since the relevances of documents were unknown before merging. The run *muena* presented in Table 9 was produced from the five individual ranked lists of documents in a round-robin fashion. The run *muenb* shown in Table 9 was produced by ranking the documents by the un-normalized relevance probabilities after combining the individual runs. And the run *muenc* shown in Table 9 was produced in the same way except that the relevance probabilities were normalized before merging. For each topic, the relevance probabilities of the documents were divided by the relevance probability of the highest-ranked document for the same topic. The simplest *raw-score* outperformed both the *normalized-score* and the *round-robin* merging strategies. We did two things to make the relevance probabilities of documents from different language collections comparable to each other. First, after concatenating the topic translations from two machine translation systems, we reduced the term frequencies by half so that the translated topics are close to the source English topics in length. Second, in query expansion, we took the same number of terms (i.e., 10) from the same number of top-ranked documents (i.e., 10) after the initial search for all five individual runs that were used to produce the multilingual runs.

There are 8 English topics and 1 Italian topic that do not have any relevant documents in the CLEF 2002 test set. For 7 out of these 9 topics, the precision using raw-score merging is better than that using round-robin or normalized merging methods. For the other two topics with no relevant documents, precision values of using the three merging methods are very close. Topic 127 entitled "Escape of Roldán" has 327 relevant documents combined for all five document languages, however, 255 relevant documents are found in the Spanish collection, 19 in the French collection, 37 in the German collection, 16 in the Italian collection, and none in the English collection. The precision for this topic is 0.6207 using the raw-score method, 0.5701 using the normalized-score method, and only 0.2851 using the round-robin method. As one would expect, when the distribution of the relevant documents across languages is highly skewed, the round-robin method would not work well since the same number of documents are taken for each language.

The p-value of the two-sided t test on comparing the raw-score and normalized-score merging methods is 0.0027, and the p-value of the t test on comparing the raw-score and round-robin merging methods is 0.0005. Our null hypotheses are that the raw-score and normalize-score methods are equally effective, and that the raw-score and round-robin methods are also equally effective. We can conclude that the performance of the raw-score merging method is significantly better than both the round-robin and the normalized-score merging methods when the translated queries have approximately the same lengths as the source queries and the same number of terms are selected for query expansion for

all monolingual runs before the merging. If the translated queries are much longer than the source queries, for example, as a result of keeping multiple translations for a single source word, or the numbers of terms selected for query expansion in the monolingual runs are very different, then the raw scores (i.e., the estimated probabilities of relevance) from the individual monolingual runs, one for each document language, may be not comparable, thus the conclusion that the raw-score merging method is significantly better than round-robin and normalized-score methods may not be valid.

8.5.2. Comparing MLIR approaches. In this section we present the evaluation results for four different approaches to multilingual information retrieval:

1. separate indexes for both topics and documents (SITD),
2. unified indexes for both topics and documents (UITD),
3. separate indexes for topics, but unified index for documents (SITUID), and
4. approximate but fast document translation (FDT).

The first MLIR method named *SITD* creates a separate index for each document language, and a separate index for each topic language. Consider the case where English topics are searched against a collection of documents in English, French, German, Italian, and Spanish as in the multilingual retrieval task in both CLEF 2001 and CLEF 2002. First, we create a separate index for each of the five document languages. Second, we translate the English topics into French, German, Italian, and Spanish, then create an index for the original English topics, and a separate index for the translated topics for each language. Third, a monolingual retrieval is performed for each of the five languages, and then the five ranked lists of retrieved documents are merged to produce a unified ranked list of documents in all languages. The SITD method uses the simple *raw-score* merging approach described in the previous section.

The second MLIR method named *UITD* differs from the first method in three ways. First, documents are indexed together to create a unified index but language-dependent stoplists and stemmers are applied in indexing. Second, the topics in the source language and all translated topics are concatenated by topic to create multilingual topics, and then the multilingual topics are indexed together to create a unified topic index. Third, the unified queries are searched directly against the unified document index to produce the ranked list of documents in all document languages. Merging of ranked lists is not needed for this method. Gey et al. (1999) originally applied this method to multilingual information retrieval.

The third MLIR method named *SITUID* creates a unified index for documents of all languages, but separate indexes for topics by language after translating the source topics into the document languages. For example, if the source language is English, and the documents languages are English, French, German, Italian, and Spanish, then one unified index is created for documents in all five languages, and one index created for the English topics only. After translating the English topics into French, German, Italian, and Spanish, one index is created for the translated topics for each language. A monolingual retrieval is carried out against the unified document index for the topics in the source language, and one for each one of the translated topic sets. The ranked lists of documents resulted from the monolingual retrieval runs are merged to produce the final ranked list. Again the raw-score

Table 10. Multilingual retrieval performances for different retrieval methods.

Run id	Test collection	Topic field	Retrieval method	Merging method	Overall recall	Average precision
muen1	CLEF 2001	T, D	SITD	Raw-score	6217/8138	0.3760
muen2	CLEF 2001	T, D	UITD	None	5399/8138	0.3156
muen3	CLEF 2001	T, D	SITUID	Raw-score	6167/8138	0.3758
muen4	CLEF 2001	T, D	FDT	None	5752/8138	0.3688
muen5	CLEF 2002	T, D	SITD	Raw-score	5880/8068	0.3762
muen6	CLEF 2002	T, D	UITD	None	4596/8068	0.2811
muen7	CLEF 2002	T, D	SITUID	Raw-score	5823/8068	0.3606
muen8	CLEF 2002	T, D	FDT	None	5580/8068	0.3800

merging method is used. The ranked lists may overlap with each other since all ranked lists are produced from the same unified document index.

The fourth MLIR method named *FDT* translates the documents word-by-word into the source topic language using bilingual wordlists created by translating document words to the topic language using MT systems. All the translated documents in the source topic language are indexed together to create one index. Then the topics in the source language are searched against the document index to directly produce a ranked list of documents. Like the second method, there is no need for merging.

Table 10 presents the evaluation results of four different approaches to multilingual retrieval on the CLEF 2001 and CLEF 2002 test collections. The source topics are in English, and only the *title* and *description* fields were indexed. The multilingual run named *muen2* was described in Section 8.5.1. All five runs before merging were produced using query expansion. The multilingual run named *muen5* was produced just like *muen1* except that the CLEF 2002 test collection was used. The *muen4* and *muen8* multilingual runs were produced using the unified document index created from the English documents and the English translations of the documents in the other four languages. The document translation from the other four languages into English was done using the approximate but very efficient method as described in Section 5. The *L&H* translator was used to translate document words in French, German, Italian and Spanish into English. For runs *muen4* and *muen8*, 10 terms were selected from 10 top-ranked documents after the initial search for query expansion. The average precision values for runs *muen4* and *muen8* before query expansion are 0.3200 and 0.3262, respectively, and the overall recall values are 5425/8138 and 4901/8068, respectively. For *muen2* and *muen6* runs, 30 terms were selected from top-ranked 20 documents for query expansion since the unified queries are about five times as long as the queries for a single language. Table 10 shows the performance of approximate document translation is comparable to query translations. The UITD approach is substantially inferior to the other three MLIR approaches.

The p-values of the t tests on comparing muen5 and muen6, muen7 and muen6, muen8 and muen6 are all much smaller than 0.05, so the SITD, SITUID, and FDT methods are significantly better than UITD method. The p-value of the t test on muen5 and muen7 is

Table 11. Fusion of two multilingual runs.

Run id	Test collections	Topic fields	Source runs	Merging method	Overall recall	Average precision
muen9	CLEF 2001	T, D	muen1 (0.3760), muen4 (0.3688)	Raw-score	6328/8138	0.4110
muen10	CLEF 2002	T, D	muen5 (0.3762), muen8 (0.3800)	Raw-score	6242/8068	0.4197

0.0796, the p-value on muen5 and muen8 is 0.8071, and the p-value on muen7 and muen8 is 0.1813. Since all three p-values are larger than 0.05, we conclude that there is no significant differences in retrieval performances among the SITD, SITUID, and FDT methods.

Braschler et al. (2002) compared query translation with document translation. In their experiments, the topics were in German, and the English, French, Italian, and Spanish documents were translated into German using MT systems. Their results show that document translation performed better than query translation in multilingual retrieval.

8.5.3. Fusion of multilingual runs. The previous section presented four different approaches to multilingual retrieval. This section evaluates the performance of simple merging of multilingual retrieval runs. Table 11 presents the results of two multilingual runs named *muen9* and *muen10*, respectively. Both runs were produced by merging two multilingual runs using the same *raw-score* merging method. The run *muen9* was the result of combining the multilingual runs *muen1* and *muen4*; and *muen10* the result of combining the multilingual runs *muen5* and *muen8*. The average precision for *muen9* run is 0.4110, while the average precision values for *muen1* and *muen4* are 0.3760 and 0.3688, respectively, as presented in Table 10. The performance of the *muen9* is 9.31% better than the best of the *muen1* and *muen4*. The average precision for *muen10* is 0.4197, while the average precision values for *muen5* and *muen8* are 0.3762 and 0.3800, respectively. The performance of *muen10* is 10.45% better than the best of *muen5* and *muen8*.

Both the p-value of the t test on comparing muen10 and muen5 and the p-value on comparing muen10 and muen6 are much smaller than 0.05, so are the p-value (0.0076) on comparing muen9 and muen1 and the p-value (0.0014) on comparing muen9 and muen4. The performance of combining two runs, one based on query translation and the other on document translation, is significantly better than the performance of either one. On the CLEF 2002 test collection, the muen5 run based on query translation outperformed the muen8 run based on document translation for 24 out of the 50 topics, for the other 26 topics, muen8 performed better. For some of the topics, the precision in the combined run muen10 is smaller than the highest precision for the same topic in run muen5 or muen8, however, the precision in the combined run muen10, for each of the 50 topics, is larger than the lowest precision for the same topic in run muen5 or muen8. On the CLEF 2001 test collection, the muen1 run based on query translation performed better than muen4 based on document translation for 27 out of 50 topics, the muen4 performed better for the other 23 topics. It is also true that the precision in the combined run muen9 is better than the lowest precision for the same topic in run muen1 or muen4. The precision values for topic 61 entitled "Siberian Oil Catastrophe" are 0.1824 for the run based on query-translation and 0.6157 for the run based on document-translation.

Braschler et al. (2002) investigated combining a query translation-based multilingual run with a document translation-based multilingual run. Their results show that the combined multilingual run was much better than the best of the two individual runs.

8.5.4. Optimal ranking. This section presents a procedure for computing the optimal performance that could possibly be achieved under the constraint that the relative ranking of the documents in the individual ranked lists is preserved. This procedure assumes that the relevances of documents are known, thus it is not useful for predicting ranks of documents in the final ranked list for multilingual retrieval. However, knowing the upper-bound performance for a set of ranked lists of documents is useful in measuring the performance of different merging strategies. We will use an example to explain the procedure. Let us assume we are going to merge three runs labeled A, B and C, as shown in Table 12. We want to find a combined ranked list such that the average precision is maximized without changing the relative rank order of the documents on the same ranked list. First we transform the individual runs shown in Table 12 into the form shown in Table 13 by grouping the consecutive irrelevant and relevant documents. Each entry in Table 13 has the form $(m, n)\{d_i, d_{i+1}, \ldots, d_j\}$, where d_i is the rank of the document ranked in the ith position in the original ranking. $\{d_i, d_{i+1}, \ldots, d_j\}$ denotes a set of consecutive irrelevant and relevant documents ranked in positions from i to j, inclusive. m is the number of irrelevant documents in the set, and n is the number of relevant documents in the set. For example, the entry (2,1) $\{B_1, B_2, B_3\}$ means the set $\{B_1, B_2, B_3\}$ has two irrelevant documents, B_1 and B_2, and one relevant document, B_3. After the transformation, the procedure can be implemented in four steps.

Step 1: Let the *active* set consist of the first set in the individual lists that contains at least one relevant document. For the example presented in Table 13, the initial *active* set is $\{(0,1)\ \{A_1\}, (2,1)\ \{B_1,B_2,B_3\}, (1,3)\ \{C_1,C_2,C_3,C_4\}\}$

Table 12. Three ranked lists of documents.

Rank	Run A	Run B	Run C
1	A_1^*	B_1	C_1
2	A_2	B_2	C_2^*
3	A_3^*	B_3^*	C_3^*
4	A_4	B_4	C_4^*

Relevant documents are marked with '*'.

Table 13. Ranked lists after transformation.

Set	Run A	Run B	Run C
1	(0, 1) $\{A_1\}$	(2, 1) $\{B_1, B_2, B_3\}$	(1,3) $\{C_1, C_2, C_3, C_4\}$
2	(1, 1) $\{A_2, A_3\}$	(1, 0) $\{B_4\}$	
3	(1, 0) $\{A_4\}$		

Table 14. Optimal ranking.

Set	Optimal ranking
1	$(0, 1)\ \{A_1\}$
2	$(1, 3)\ \{C_1, C_2, C_3, C_4\}$
3	$(1, 1)\ \{A_2, A_3\}$
4	$(2, 1)\ \{B_1, B_2, B_3\}$
5	$(1, 0)\ \{A_4\}$
6	$(1, 0)\ \{B_4\}$

Step 2: Choose the element in the *active* set with the smallest number of irrelevant documents. If there are two or more elements with the smallest number of irrelevant documents, then choose the element that also contains the largest number of relevant documents. If there are two or more elements with the same smallest number of irrelevant documents and the same largest number of relevant documents in the current *active* set, then randomly choose one of them. Append the selected element to the final ranked list. If the next set appearing immediately after the selected element contains at least one relevant document, then add the next set to the current *active* set. That is, sort the *active* set by m as the major order in increasing order, and by n as the minor order in decreasing order, then take out the first element and put it in the final ranked list.

Step 3: Repeat Step 2 until the current *active* set is empty.

Step 4: If the final ranked list has fewer than 1000 documents, append more irrelevant documents drawn from any individual list to the final ranked list.

The optimal ranking after reordering the sets is presented in Table 14. The optimal average precision by combining the five monolingual runs that were used in producing the muen5 run is 0.5177 with overall recall of 6392/8068. The performances of the raw-score and score-normalizing merging are far below the upper-bound performance that could possibly be achieved.

9. Conclusions

The goal of this paper has been to present, within a context of the CLEF performance evaluation format, the constituent components making up successful cross-language information retrieval.

We have presented a technique for incorporating blind relevance feedback into a document ranking formula based on logistic regression analysis. Query expansion was implemented in three steps: first, a pre-specified number of terms are selected from a given number of top-ranked documents after the initial search; second, weights are assigned to the selected terms; third, the selected terms are combined with the initial query terms to create an expanded query which is used to generate the final retrieval results. Five methods for term selection were evaluated over ten experiments in five different langages (English, French, German, Italian, and Spanish). The improvement in average precision brought by query expansion via blind relevance feedback ranges from 4.69% to 19.42% for monolingual retrieval run,

No single term selection method showed consistently better performance. Query expansion improved from 10.85% to 26.70% in bilingual retrieval experiments.

We have presented a procedure to decompose German compound words and discussed the advantages obtained from such decompounding. A base dictionary consisting of non-compound words is used to decompose a compound word. When there are multiple ways to decompose a compound, a selection rule is invoked to choose the most appropriate way to split a compound. German decompounding improved the average precision of German monolingual retrieval by 11.47%. Decompounding increased the average precision for bilingual retrieval to German from English or French, with increases ranging from 8.4% to 11.46%. In summary, both blind relevance feedback and decompounding in German have been shown to be effective in monolingual and bilingual retrieval. The amount of improvement of performance by decompounding varies from one set of topics to another.

A set of experiments evaluated combinations of monolingual, language-specific stemming in combination with blind feedback query expansion and German term decompounding. The interesting part of the results is that these three techniques are complementary—each component enhances the overall performance, which, when all techniques are combined, can result in a fifty percent improvement in average precision over the query set.

Three different merging strategies in multilingual retrieval were evaluated. The simplest *raw-score* merging strategy worked better than the *normalized-score* strategy, but both outperformed *round-robin* merging. To make the relevance scores of the documents from different collections as closely comparable as possible, we selected the same number of terms from the same number of top-ranked documents after the initial search for query expansion in all the runs that were combined to produce the unified ranked lists of documents in multiple languages. We used two machine translation systems to translate English topics to French, German, Italian and Spanish, and combined by topic the translations from the two machine translation systems. We reduced the term frequencies in the combined translated topics by half so that the combined translated topics closely approximate the length of the source English topics.

We presented an algorithm for generating the optimal ranked list of documents when the document relevances are known. The optimal performance can then be used to measure the performances of different merging strategies.

An approximate but fast document translation method based on MT systems was proposed and evaluated in multilingual retrieval. In this method, all documents are translated, word-by-word, into the source language using MT-induced lexicons.

For multilingual retrieval, the fast document translation-based approach is as effective as the query translation-based one. When a fast document translation-based multilingual retrieval run was combined with a query translation-based multilingual retrieval run, the performance of the combined multilingual run was about 10% better than that of the best individual run.

Acknowledgments

The authors would like to thank the anonymous reviewers, Carol Peters, and Martin Braschler for many constructive comments. They would also like to thank Michael Buckland for his

comments on the draft manuscript. This research was supported by DARPA under research grant N66001-00-1-8911 as part of the DARPA Translingual Information Detection, Extraction, and Summarization Program (TIDES).

References

Ballesteros L and Croft WB (1997) Phrasal translation and query expansion techniques for cross-language information retrieval. In: Proceedings of the SIGIR'97. The ACM Press, New York, pp. 84–91.

Ballesteros L and Croft W (1998) Statistical methods for cross-language information retrieval. In: Grefenstette G, Ed. Cross Language Information Retrieval, Kluwer.

Braschler M, Ripplinger B and Schäuble P (2002) Experiments with the eurospider retrieval system for CLEF 2001. In: Peters C et al, Eds. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2406. Springer-Verlag, Berlin, pp. 102–110.

Callan JP, Lu Z and Croft WB (1995) Searching distributed collections with inference networks. In: Proceedings of the ACM-SIGIR. The ACM Press, New York, pp. 21–28.

Chen A (2002a) Multilingual information retrieval using English and Chinese queries. In: Peters C et al, Eds. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2406. Springer-Verlag, Berlin, pp. 44–58.

Chen A (2002b) Cross-language retrieval experiments at CLEF 2002. In: Peters C, Ed. Working Notes for the CLEF 2002 Workshop 19–20 Sept., Rome, Italy, pp. 5–20.

Cooper WS, Chen A and Gey FC (1994) Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In: Harman DK, Ed. The Second Text REtrieval Conference (TREC-2), pp. 57–64.

Gey FC, Jiang H, Chen A and Larson RR (1999) Manual queries and machine translation in cross-language retrieval and interactive retrieval with Cheshire II at TREC-7. In: Voorhees EM and Harman DK, Eds. The Seventh Text REtrieval Conference (TREC-7). NIST Special Publication 500-24, National Institute of Standards and Technology, Gaithersburg, MD, pp. 527–540.

Gey FC, Jiang H, Petras V and Chen A (2001) Cross-language retrieval for the CLEF collections—Comparing multiple methods of retrieval. In: Peters C, Ed. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2069. Springer-Verlag, Berlin, pp. 116–128.

Grefenstette G (1998), Ed. Cross-Language Information Retrieval. Kluwer Academic Publishers, Boston, MA.

Harman D (1992) Relevance feedback and other query modification techniques. In: Frakes W and Baeza-Yates R, Eds. Information Retrieval: Data Structures & Algorithms. Prentice Hall, pp. 241–263.

Hiemstra D, Kraaij W, Pohlmann R and Westerveld T (2001) Translation resources, merging strategies, and relevance feedback for cross-language information retrieval. In: Peters C, Ed. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2069. Springer-Verlag, Berlin, pp. 102–115.

Hull D (1993) Using statistical testing in the evaluation of retrieval experiments, In: Proceedings of the SIGIR'93. The ACM Press, New York, pp. 329–338.

Kraaij W (2002) TNO at CLEF 2001: Comparing translation resources. In: Peters C et al., Eds. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2406. Springer-Verlag, Berlin, pp. 78–93.

Lezius W, Rapp R and Wettler M (1988) A freely available morphological analyzer, disambiguator and context sensitive Lemmatizer for German. In: COLING-ACL'98, pp. 743–748.

McCarley JS and Roukos S (1998) Fast document translation for cross-language information retrieval. In: Farwell D, Gerber L and Hovy E, Eds. Machine Translation and the Information Soup. Lecture Notes in Computer Science, Vol. 1529. Springer-Verlag, Berlin, pp. 150–157.

McNamee P and Mayfield J (2002) JHU/APL experiments at CLEF: Translation resources and score normalization. In: Peters C et al., Eds. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2406. Springer-Verlag, Berlin, pp. 193–208.

Monz C and de Rijke M (2002) Shallow morphological analysis in monolingual information retrieval for Dutch, German, and Italian. In: Peters C et al., Eds. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2406, Springer-Verlag, Berlin, pp. 263–277.

Oard DW and Diekema AR (1998) Cross-language information retrieval. Annual Review of Information Science and Technology, 33:223–256.

Oard DW (1998) A comparative study of query and document translation for cross-language information retrieval. In: Farwell D, Gerber L and Hovy E, Eds. Machine Translation and the Information Soup. Lecture Notes in Computer Science, Vol. 1529. Springer-Verlag, Berlin, pp. 472–483.

Peters C (2001), Ed. Evaluation of cross-language information retrieval systems. Lecture Notes in Computer Science, Vol. 2069, Springer-Verlag, Berlin.

Peters C, Braschler M, Gonzalo J and Kluck M (2002a), Eds. Evaluation of cross-language information retrieval systems. Lecture Notes in Computer Science, Vol. 2406, Springer-Verlag, Berlin.

Peters C (2002b), Ed. Working Notes for the CLEF 2002 Workshop 19–20 Sept., Rome, Italy.

Picchi E and Peters C (1998) Cross language information retrieval: A system for comparable corpus querying. In: Grefenstette G, Ed. Cross Language Information Retrieval, Kluwer.

Pirkila A, Hedlund T, Keskustalo H and Jävelin K (2001) Dictionary-based cross-language information retrieval: Problems, methods, and research findings. Information Retrieval, 4:209–230.

Porter M (2001) Snowball: A language for stemming algorithms. Available at http://snowball.tartarus.org/texts/introduction.html.

Rice JA (1995) Mathematical Statistics and Data Analysis, 2nd edn. Duxbury Press, Belmont, California.

Robertson SE and Sparck Jones K (1976) Relevance weighting of search terms. Journal of the American Society for Information Science, 129–146.

Robertson SE, Walker S and Beaulieu M (2000) Experimentation as a way of life: Okapi at TREC. Information Processing & Management, 36:95–108.

Savoy J (2002a) Report on CLEF 2001 experiments: Effective combined query-translation approach. In: Peters C, Ed. Evaluation of Cross-Language Information Retrieval Systems. Lecture Notes in Computer Science, Vol. 2069. Springer-Verlag, Berlin, 2001, pp. 27–43.

Savoy J (2002b) Report on CLEF 2002 experiments: Combining multiple sources of evidence. In: Peters C, Ed. Working Notes for the CLEF 2002 Workshop 19–20 Sept., Rome, Italy, pp. 31–46.

Voorhees EM and Harman DK (1998), Eds. The Seventh Text Retrieval Conference (TREC-7). NIST.

Voorhees EM and Harman DK (1999), Eds. The Eighth Text Retrieval Conference (TREC-8). NIST.

Yang Y, Carbonell J, Brown R and Frederking R (1998) Translingual information retrieval: Learning from bilingual corpora. Artificial Intelligence, 103:323–345.

Xu J, Weischedel R and Fraser A (2001) TREC-9 cross-lingual retrieval at BBN. In: Voorhees EM and Harman DK, Eds. The Ninth Text Retrieval Conference (TREC-9), NIST Special Publication 500-249, pp. 106–116.

Xu J, Weischedel R and Fraser A (2002) Trec 2001 cross-lingual retrieval at BBN. In: Voorhees EM and Harman DK, Eds. The Tenth Text Retrieval Conference (TREC-2001), NIST Special Publication 500-250, pp. 68–77.