Language-dependent and Language-independent Approaches to Cross-Lingual Text Retrieval

Jaap Kamps, Christof Monz, Maarten de Rijke, and Börkur Sigurbjörnsson

Language & Inference Technology Group, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands
E-mail: {kamps, christof, mdr, borkur}@science.uva.nl

Abstract. We investigates the effectiveness of *language-dependent* approaches to document retrieval, such as stemming and decompounding, and constrast them with *language-independent* approaches, such as character n-gramming. In order to reap the benefits of more than one type of approach, we also consider the effectiveness of the combination of both types of approaches. We focus on document retrieval in nine European languages: Dutch, English, Finnish, French, German, Italian, Russian, Spanish, and Swedish. We look at four different cross-lingual information retrieval tasks: monolingual, bilingual, multilingual, and domain-specific retrieval. The experimental evidence is obtained using the 2003 test suite of the cross-language evaluation forum (CLEF).

1 Introduction

Researchers in Information Retrieval (IR) have experimented with a great variety of approaches to document retrieval for European languages. Differences between these approaches range from the text representation used (e.g., whether to apply morphological normalization or not, or which type of query formulation to use), to the choice of search strategy (e.g., which weighting scheme to use, or whether to use blind feedback). We focus on approaches using different document representations, but using the same retrieval settings and weighting scheme. In particular, we focus on different approaches to morphological normalization or tokenization. We conducted experiments on nine European languages (Dutch, English, Finnish, French, German, Italian, Russian, Spanish, and Swedish). There are notable differences between these languages, such as the complexity of inflectional and derivational morphology [1].

A recent overview of monolingual document retrieval can be found in [2]. The options considered in [2] include word-based runs (indexing the tokens as they occur in the documents), stemming (using stemmers from the Snowball family of stemming algorithms), lemmatizing (using the lemmatizer built into the TreeTagger part-of-speech tagger), and compound splitting (for compound forming languages such as Dutch, Finnish, German, and Swedish). Additionally, there are experiments with adding character n-grams (of length 4 and 5). The main lessons learned in [2] were two fold. First, there is no language for which the best performing run significantly improves over the "compound split and

stem" run (treating splitting as a no-op for non-compound forming languages). Second, the hypothesis that adding 4-gramming is the best strategy is refuted for Spanish only. Notice that these comparisons did not involve combinations of runs, but only runs based on a single index.

The aim of this paper is to redo some of the experiments of [2], to investigate the combination of approaches, and to extend these experiments to a number of cross-lingual retrieval tasks (we give details below). In particular, we will investigate the effectiveness of *language-dependent* approaches to document retrieval, i.e., approaches that require detailed knowledge of the particular language at hand. The best known example of a language-dependent approach is the use of stemming algorithms. The effectiveness of stemming in English is a recurring issue in a number of studies [3,4]. Here we consider the effectiveness of stemming for nine European languages. Another example of a language-dependent approach is the use of decompounding strategies for compound-rich European languages, such as Dutch and German [5]. Compounds formed by the concatenation of words are rare in English, although exceptions like *database* exist. We will also investigate the effectiveness of *language-independent* approaches to document retrieval, i.e., approaches that do not depend on knowledge of the language at hand. The best known example of language-independent approaches is the use of character n-gramming techniques. Finally, we will investigate whether both approaches to document retrieval can be fruitfully combined [6]. Hoping to establish the robustness and effectiveness of these approaches for a whole range of cross-lingual retrieval tasks, we supplement the monolingual retrieval experiments with bilingual retrieval experiments, with multilingual experiments, and with domain-specific experiments. Experimental evaluation is done on the test suite of the Cross-Language Evaluation Forum [7].

The paper is organized as follows. In Section 2 we describe the FlexIR system as well as the approaches used for all of the crosslingual retrieval tasks. In Section 3 we discuss our experiments for monolingual retrieval (in Section 3.1), bilingual retrieval (in Section 3.2), multilingual retrieval (in Section 3.3), and domain-specific retrieval (in Section 3.4). Finally, in Section 4, we offer some conclusions drawns from our experiments.

2 System Description

2.1 Retrieval Approach.

All retrieval runs used FlexIR, an information retrieval system developed at the University of Amsterdam [5]. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl and supports many types of preprocessing, scoring, indexing, and retrieval tools.

Retrieval Model. FlexIR supports several retrieval models, including the standard vector space model, language models, and probabilistic models. All runs

reported in the paper use the vector space model with the Lnu.ltc weighting scheme [8] to compute the similarity between a query and a document. For all the experiments, we fixed *slope* at 0.2; the pivot was set to the average number of unique words per document.

Morphological Normalization. We apply a range of language-dependent and language-independent approaches to morphological normalization or tokenization.

Words — We consider as a baseline the straightforward indexing of the words as encountered in the collection. We do some limited sanitizing: diacritics are mapped to the unmarked character, and all characters are put in lower-case. Thus a string like '`Information Retrieval`' is indexed as '`information retrieval`' and a string like the German '`Raststätte`' (English: motorway restaurant) is indexed as '`raststatte`.'

Stemming — The stemming or lemmatization of words is the most popular language-dependent approach to document retrieval. We use the set of stemmers implemented in the Snowball language [9]. Thus a string like '`Information Retrieval`' is indexed as the stems '`inform retriev`.'

An overview of stemming algorithms can be found in [10]. The string processing language Snowball is specifically designed for creating stemming algorithms for use in Information Retrieval. It is partly based on the familiar Porter stemmer for English [11], and provides stemming algorithms for all the nine European languages that we consider in this paper. We perform the same sanitizing operations as for the word-based run.

Decompounding — For the compound rich languages, Dutch, German, Finnish, and Swedish, we apply a decompounding algorithm. We treat all words occurring in the CLEF corpus as potential base words for decompounding, and also use their associated collection frequencies. We ignore words of length less than four characters as potential compound parts, thus a compound must consist of at least eight characters. As a safeguard against oversplitting, we only regard compound parts that have a higher collection frequency than the compound itself. We consider linking elements -*s*-, -*e*-, and -*en*- for Dutch; -*s*-, -*n*-, -*e*-, and -*en*- for German; -*s*-, -*e*-, -*u*-, and -*o*- for Swedish; and none for Finnish. We prefer a split with no linking element over a split with a linking element, and a split with a single character linker over a two character linker.

Each document in the collection is analyzed and if a compound is identified, the compound is kept in the document and all of its parts are added to the document. Thus a string like the Dutch '`boekenkast`' (English: bookshelf) is indexed as '`boekenkast boek kast`.' Compounds occurring in a query are analyzed in a similar way: the parts are simply added to the query. Since we expand both the documents and the queries with compound parts, there is no need for compound formation [12].

n-Gramming — Character n-gramming is the most popular language-independent approach to document retrieval. Our n-grams were not allowed to cross word boundaries. This means that the string '`Information Retrieval`' is indexed as the fourteen 4-gram tokens '`info nfor form orma rmat mati atio tion retr etri trie riev ieva eval`'. We experimented with two n-gram approaches. First, we replaced the words with their n-grams. Second, we added the n-grams to the documents but kept the original words as well.

Character n-grams are an old technique for improving retrieval effectiveness. An excellent overview of n-gramming techniques for cross-lingual information retrieval is given in [13]. Again, we perform the same sanitizing operations as for the word-based run.

Character Encodings. Until CLEF 2003, the languages of the CLEF collections all used the Latin alphabet. The addition of the new CLEF language, Russian, is challenging because of the use of a non-Latin alphabet. The Cyrillic characters used in Russian can appear in a variety of font encodings. The collection and topics are encoded using the UTF-8 or Unicode character encoding. We converted the UTF-8 encoding into a 1-byte per character encoding KOI8 or KOI8-R (for *Kod Obmena Informatsii* or Code of Information Exchange).[1] We did all our processing, such as lower-casing, stopping, stemming, and n-gramming, on documents and queries in this KOI8 encoding. Finally, to ensure the proper indexing of the documents using our standard architecture, we converted the resulting documents into the Latin alphabet using the Volapuk transliteration. We processed the Russian queries similar to the documents.

Stopwords. Both topics and documents were stopped using the stopword lists from the Snowball stemming tool [9], for Finnish we used the Neuchâtel-stoplist [14]. Additionally, we removed topic specific phrases such as 'Find documents that discuss ...' from the queries. We did not use a stop stem or n-gram list, but we first used a stop *word* list, and then stemmed/n-grammed the topics and documents.

Blind Feedback. Blind feedback was applied to expand the original query with related terms. Term weights were recomputed by using the standard Rocchio method [15], where we considered the top 10 documents to be relevant and the bottom 500 documents to be non-relevant. We allowed at most 20 terms to be added to the original query.

Combination Methods. For each of the CLEF 2003 languages we created base runs using a variety of indexing methods (see below). We then combined these base runs using one of two methods, either a weighted or an unweighted

[1] We used the excellent Perl package `Convert::Cyrillic` for conversion between character encodings and for lower-casing Cyrillic characters.

combination. An extensive overview of combination methods for cross-lingual information retrieval is given in [16].

The weighted combination was produced as follows. First, we normalized the retrieval status values (RSVs), since different runs may have radically different RSVs. For each run we reranked these values in $[0, 1]$ using:

$$RSV_i' = \frac{RSV_i - min_i}{max_i - min_i};$$

this is the Min_Max_Norm considered in [17]. Next, we assigned new weights to the documents using a linear interpolation factor λ representing the relative weight of a run:

$$RSV_{new} = \lambda \cdot RSV_1 + (1 - \lambda) \cdot RSV_2.$$

For $\lambda = 0.5$ this is similar to the simple (but effective) combSUM function used by Fox and Shaw [18]. The interpolation factors λ were obtained from experiments on the CLEF 2002 data sets (whenever available). When we combine more than two runs, we give all runs the same relative weight, resulting effectively in the familiar combSUM method.

Statistical Significance. Finally, to determine whether the observed differences between two retrieval approaches are statistically significant, we used the bootstrap method, a non-parametric inference test [19,20]. We take 100,000 resamples, and look for significant improvements (one-tailed) at significance levels of 0.95 (*); 0.99 (**); and 0.999 (***).

3 Experiments

In this section, we describe our experiments for the monolingual task, the bilingual task, the multilingual task, and the domain-specific task.

3.1 Monolingual Retrieval

For the monolingual task, we conducted experiments with a number of language-dependent and language-independent approaches to document retrieval. All our monolingual runs used the title and description fields of the topics.

Baseline. Our baseline run is straightforwardly indexing the words as encountered in the collection (with case-folding and mapping marked characters to the unmarked symbol). The mean-average-precision (MAP) scores are shown in Table 1. The baseline run is fairly high performing run for most languages. In particular, Dutch with a MAP of 0.4800 performs relatively well.

Table 1. Word-based run.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485

Table 2. Snowball stemming algorithm.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485
Stems	0.4652	0.4273	0.3998	0.4511	0.4504	0.4726	0.2536	0.4678	0.3707
%Ch.	-3.1	-4.7	+25.9	+4.6	+19.0	+2.1	-0.6	+6.2	+6.4
Stat.	-	-	*	-	***	-	-	*	-

Stemming. For all eight languages, we use a stemming algorithm from the Snowball family [9] (see Section 2). The results are shown in Table 2. The results are mixed. On the one hand, we see a decrease in retrieval effectiveness for Dutch, English, and Russian. On the other hand, we see an increase in retrieval effectiveness for Finnish, French, German, Italian, Spanish, and Swedish. The improvements for Finnish, German, and Spanish are statistically significant.

Decompounding. Compounds are split using the method described in Section 2. We decompound documents and queries for the four compound-rich languages: Dutch, Finnish, German, and Swedish. After decompounding, we apply the same stemming procedure as above. The results are shown in Table 3. The

Table 3. Decompounding.

	Dutch	*Finnish*	*German*	*Swedish*
Words	0.4800	0.3175	0.3785	0.3485
Split+Stem	0.4984	0.4453	0.4840	0.3957
%Ch.	+3.8	+40.3	+27.9	+13.5
Stat.	-	***	***	-

results for decompounding are positive overall. We now see an improvement for Dutch, and further improvement for Finnish, German, and Swedish.

Our results indicate that for all four compound forming languages, Dutch, Finnish, German, and Swedish, we should decompound before stemming. We treat the resulting (compound-split and) stem runs as a single language-dependent approach, where we only decompound the four compound-rich languages. The results are shown in Table 4. These resulting (compound-split and) stem runs improve for all languages, except for English and the low-performing Russian.

n-Gramming. Both topic and document words are n-grammed, using the settings discussed in Section 2. For all languages we use 4-grams, that is, character

Table 4. (Compound splitting and) stemming algorithms.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485
Split+Stem	0.4984	0.4273	0.4453	0.4511	0.4840	0.4726	0.2536	0.4678	0.3957
%Ch.	+3.8	-4.7	+40.3	+4.6	+27.9	+2.1	-0.6	+6.2	+13.5
Stat.	-	-	⋆⋆⋆	-	⋆⋆⋆	-	-	⋆	-

n-grams of length 4. The results for replacing the words with n-grams are shown

Table 5. 4-Gramming.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485
4-Grams	0.4488	0.3731	0.4676	0.4142	0.4639	0.3883	0.2871	0.4545	0.3751
%Ch.	-6.5	-16.8	+47.3	-4.0	+22.6	-16.2	+12.5	+3.2	+7.6
Stat.	-	⋆⋆	⋆⋆	-	⋆⋆	⋆⋆	-	-	-

in Table 5. We see a decrease in performance for four languages: Dutch, English, French, and Italian, and an improvement for the other five languages: Finnish, German, Russian, Spanish, and Swedish. The increase in retrieval effectiveness is statistically significant for Finnish and German, the decrease in performance is significant for English and Italian. The results are mixed, and the technique of character n-gramming is far from being a panacea.

We explore a second language-independent approach, by adding the n-grams to the free-text of the documents, rather than replacing the free-text with n-grams. The results of adding n-grams are shown in Table 6. The runs improve

Table 6. 4-Gramming while retaining words.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485
Word+4-Gr.	0.4996	0.4119	0.4905	0.4616	0.5005	0.4227	0.3030	0.4733	0.4187
%Ch.	+4.1	-8.1	+54.5	+7.0	+32.2	-8.7	+18.8	+7.4	+20.1
Stat.	-	⋆	⋆⋆⋆	-	⋆⋆⋆	-	⋆	⋆	⋆

over pure n-grams for all the nine languages. With respect to the words baseline, we see a decrease in performance for English and Italian, and an improvement for the other seven languages: Dutch, Finnish, French, German, Russian, Spanish, and Swedish. The deviating behavior for Italian may be due to the different ways of encoding marked characters in the Italian sub-collections [7]. Improvements are significant for five of the languages, namely Finnish, German, Russian, Spanish, and Swedish. However, the decrease in performance for English remains significant too.

Combining. It is clear from the results above that there is no equivocal best strategy for monolingual document retrieval. For English, our baseline run scores best. For Italian, the stemmed run scores best. For the other seven languages, Word+4-Gramming scores best. Here, we consider the combination of language-dependent and language-independent approaches to document retrieval. We apply a weighted combination method, also referred to as linear fusion. From the experiments above we select the approaches that exhibit the best overall performance:

Best language-dependent approach is to decompound for Dutch, Finnish, German, and Swedish, and then apply a stemming algorithm.

Best language-independent approach is to add n-grams while retaining the original words.

In particular, we combine the (compound split and) stem run of Table 3 with the Word+4-Gram run of Table 6. The used interpolation factors are based on experiments using the CLEF 2002 test suite (whenever available). We used the following relative weights of the n-gram run: 0.25 (Dutch), 0.4 (English), 0.51 (Finnish), 0.66 (French), 0.36 (German), 0.405 (Italian), 0.60 (Russian), 0.35 (Spanish), and 0.585 (Swedish).

Table 7. Combination of (Compound-splitting and) Stemming and adding 4-Grams.

	Dutch	*English*	*Finnish*	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.4800	0.4483	0.3175	0.4313	0.3785	0.4631	0.2551	0.4405	0.3485
Combination	0.5072	0.4575	0.5236	0.4888	0.5091	0.4781	0.2988	0.4841	0.4371
%Ch.	+5.7	+2.1	+64.9	+13.3	+34.5	+3.2	+17.1	+9.9	+25.4
Stat.	-	-	***	**	***	-	*	***	**

The results are shown in Table 7. We find only positive results: all languages improve over the baseline, even English! Even though both English runs scored lower than the baseline (one of them even significantly lower), the combination improves over the baseline. The improvements for six of the languages, Finnish, French, German, Russian, Spanish, and Swedish, are significant. All languages except Russian improve over the best run using a single index.

3.2 Bilingual Retrieval

We restrict our attention here to bilingual runs using the English topic set. All our bilingual runs used the title and description fields of the topics. We experimented with the WorldLingo machine translation [21] for translations into Dutch, French, German, Italian, and Spanish. For translation into Russian we used the PROMT-Reverso machine translation [22]. For translations into Swedish, we used the the first mentioned translation in the Babylon on-line dictionary [23]. Since we use the English topic set, the results for English are

the monolingual runs discussed above in Section 3.1. We also ignore English to Finnish retrieval for lack of an acceptable automatic translation method. Thus, we focus on seven European languages.

We created the exact same set of runs as for the monolingual retrieval task described above: a word-based baseline run; a stemmed run with decompounding for Dutch, German, and Swedish; a words+4-gram run; and a weighted combination of words+4-gram and (split and) stem runs. We use the following relative weights of the words+4-gram run: 0.6 (Dutch), 0.7 (French), 0.5 (German), 0.6 (Italian), 0.6 (Russian), 0.5 (Spanish), and 0.8 (Swedish).

Table 8. Bilingual runs using EN topic set. Best scores are in boldface. We compare the best scoring run with the word-based baseline run.

	Dutch	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Words	0.3554	0.3547	0.3378	0.3810	0.1379	0.3246	0.1187
(Split+)Stem	**0.4043**	0.3567	0.3968	0.3860	**0.2270**	0.3588	0.1898
Word+4-Grams	0.3690	0.3762	0.4228	0.3801	0.1983	0.3775	0.2371
Combination	0.3971	**0.3951**	**0.4479**	**0.3927**	0.2195	**0.3888**	**0.2478**
%Change	+13.8	+11.4	+32.6	+3.1	+64.6	+19.8	+108.8
Stat.Sign.	*	-	***	-	**	**	***

Table 8 shows our MAP scores for the English to Dutch, French, German, Italian, Russian, Spanish, and Swedish, bilingual runs. For our official runs for the 2003 bilingual task, we refer the reader to [24]. Adding 4-grams improves retrieval effectiveness over the word-based baseline for all languages except Italian (which exhibits a marginal drop in performance). The stemmed, and decompounded for Dutch, German, and Swedish, runs do improve for all seven languages. The Dutch stemmed and decompounded run and the Russian stemmed run turn out to be particularly effective, and outperform the respective n-gram and combination runs. A conclusion on the effectiveness of the Russian stemmer, based on only the monolingual evidence earlier, would prove to be premature. Although the stemmer failed to improve retrieval effectiveness for the monolingual Russian task, it is effective for the bilingual Russian task. For the other five languages (French, German, Italian, Spanish, and Swedish) the combination of stemming and n-gramming results in the best bilingual performance. The best performing run does significantly improve over the word-based baseline for five of the seven languages: Dutch, German, Russian, Spanish, and Swedish.

The results on the English topic set are, as expected, somewhat lower than the monolingual runs. Table 9 shows the decrease in effectiveness of the best bilingual run compared to the best monolingual run for the respective target language. The difference ranges from a 12% decrease (German) to a 43% decrease (Swedish) in MAP score. The big gap in performance for Swedish is most likely a result of the use of a translation dictionary, rather than a proper machine translation. The results for the other languages seem quite acceptable, considering that we used a simple, straightforward machine translation for the

Table 9. Decrease in effectiveness for bilingual runs.

	Dutch	*French*	*German*	*Italian*	*Russian*	*Spanish*	*Swedish*
Best monolingual	0.5072	0.4888	0.5091	0.4781	0.3030	0.4841	0.4371
Best bilingual	0.4043	0.3951	0.4479	0.3927	0.2270	0.3888	0.2478
%Change	−20.3	−19.2	−12.0	−17.9	−25.1	−19.7	−43.3
Stat.Sign.	⋆	-	⋆⋆⋆	-	⋆⋆	⋆⋆	⋆⋆⋆

bilingual tasks [21]. The bilingual results do, in general, confirm the results obtained for the monolingual task. This increases our confidence in the effectiveness and robustness of the language-dependent and language-independent approaches employed for building the indexes.

3.3 Multilingual Retrieval

We used the English topic set for our multilingual runs, using only the title and description fields of the topics. We use the English monolingual run (see Section 3.1) and the English to Dutch, French, German, Italian, Spanish, and Swedish bilingual runs (see Section 3.2) to construct our multilingual runs. There are two different multilingual tasks. The small multilingual task uses four languages: English, French, German, and Spanish. The large multilingual task extends this set with four additional languages: Dutch, Finnish, Italian, and Swedish. Recall from our bilingual experiments in Section 3.2 that we do not have an English to Finnish bilingual run, and that our English to Swedish bilingual runs perform somewhat lower due to the use of a translation dictionary.

This prompted the following three sets of experiments:

1. on the four languages of the small multilingual task (English, French, German, and Spanish),
2. on the six languages for which we have an acceptable machine translation (also including Dutch and Italian), and
3. on the seven languages (also including Swedish, but no Finnish documents) for which we have, at least, an acceptable bilingual dictionary.

For each of these experiments, we build a number of combined runs, where we use the unweighted combSUM rule introduced by [18]. First, we combine a single, uniform run per language, in all cases the bilingual words+4-gram run (see Section 3.1 and 3.2). Second, we again use a single run per language, the weighted combination of the words+4-gram and (Split+)Stem run (see Section 3.1 and 3.2). Third, we form a big pool of runs, two per language: the Word+4-Grams runs and the (Split+)Stem runs.

Table 10 shows our multilingual MAP scores for the small multilingual task (covering four languages) and for the large multilingual task (covering eight languages). For all multilingual experiments, first making a weighted combination per language outperforms the unweighted combination of all Word+4-Grams run and all (Split+)Stem runs. However, as we add languages, we see that the

Table 10. Overview of MAP scores for multilingual runs.

	Multi-4	*Multi-8*	
		(without FI/SV)	*(without FI)*
Word+4-Gram	0.2953	0.2425	0.2475
Combined Word+4-Gram/(Split+)Stem	0.3341	0.2806	0.2860
Both Word+n-Gram and (Split+)Stem	0.3292	0.2764	0.2843

unweighted combination of all Word+4-Grams runs and all (Split+)Stem runs performs almost as well as the weighted combinations.

Our results show that multilingual retrieval on a subpart of the collection (leaving out one or two languages) can still be an effective strategy. However, the results also indicate that the inclusion of further languages does consistently improve MAP scores.

3.4 Domain-specific Retrieval

For our domain-specific retrieval experiments, we used the *German Information Retrieval Test-database* (GIRT). We focus on monolingual experiments using the German topics and the German collection. We used the title and description fields of the topics, and used the title and abstract fields of the collection. We experimented with a reranking strategy based on the keywords assigned to the documents, the resulting rerank runs also use the controlled-vocabulary fields in the collection.

We make three different indexes mimicking the settings used for our monolingual German experiments discussed in Section 3.1. First, we make an word-based index as used in our baseline runs. Second, we make a stemmed index in which we did not use a decompounding strategy. Third, we build a Word+4-Grams index.

Table 11 contains our MAP scores for the GIRT monolingual task. The results for the GIRT tasks show the effectiveness of stemming and n-gramming approaches over a plain word index. Notice also that the performance of German domain-specific retrieval are somewhat lower than those of German monolingual retrieval.

Table 11. Overview of MAP scores for GIRT runs.

	GIRT	*%Change*	*Stat.sign.*
Words (baseline)	0.2360		
Stems	0.2832	+20.0	***
Word+4-Grams	0.3449	+46.1	***

The main aim of our domain-specific experiments is to find way to exploit the manually assigned keywords in the collection. These keywords are based

on the controlled-vocabulary thesaurus maintained by GESIS [25]. In particular, we experiment with an improved version of the keyword-based reranking strategy introduced in [6]. We calculate vectors for the keywords based on their (co)occurrences in the collection. The main innovation is in the use of higher dimensional vectors for the keywords, for which we use the best reduction onto a 100-dimensional euclidean space. The reranking strategy is as follows. We calculate vectors for all initially retrieved documents, by simply taking the mean of the vectors of keywords assigned to the documents. We calculate a vector for a topic by taking the relevance-weighted mean of the top 10 retrieved documents. We now have a vector for each of the topics, and for each of the retrieved documents. Thus, ignoring the RSV of the retrieved documents, we can simply rerank all documents by the euclidean distance between the document and topic vectors. Next, we combine the original text-based similarity scores with the keyword-based distances using the unweighted combSUM rule of [18].

The results of the reranking strategy are shown in the rest of Table 12. For

Table 12. Overview of MAP scores for GIRT runs. We compare the rerank runs with the respective orginal runs.

	GIRT baseline	*Rerank*	*%Change*	*Stat.sign.*
Words	0.2360	0.2863	+21.31%	***
Stems	0.2832	0.3361	+18.68%	***
Word+4-Grams	0.3449	0.3993	+15.77%	***

all the three index approaches, the results are positive. There is a significant improvement of retrieval effectiveness due to the keyword-based reranking method. The obtained improvement is additional to the improvement due to blind feedback, and consistent even for high performing base runs.

4 Conclusions

This paper investigated the effectiveness of language-dependent and language-independent approaches to cross-lingual text retrieval. The experiments described in this paper indicate the following. First, morphological normalization does improve retrieval effectiveness, especially for languages that have a more complex morphology than English. We also showed that n-gram-based can be a viable option in the absence of linguistic resources to support deep morphological normalization. Although no panacea, the combination of runs provides a method that may help improve base runs, even high quality base runs. The interpolation factors required for the best gain in performance seem to be fairly robust across topic sets. Moreover, the effectiveness of the unweighted combination of runs is usually close to the weighted combination, and the difference seems to diminish with the number of runs being combined. Our bilingual experiments showed that a simple machine translation strategy can be effective for bilingual

retrieval. The combination of bilingual runs, in turn, leads to an effective strategy for multilingual retrieval. Finally, our results for domain-specific retrieval show the effectiveness of stemming and n-gramming even for specialized collection. Moreover, manually assigned classification information in such scientific collections can be fruitfully exploited for improving retrieval effectiveness.

Our future research is to extend the described experiments to other retrieval models. In particular, we are considering the Okapi weighting scheme [26], and a language model [27]. We have started conducting initial experiments using these alternative retrieval models. In [24], we reported on Okapi and language model runs using the (decompounded and) stemmed indexes for Dutch, German, Spanish, and Swedish. In fact, these combinations of different retrieval models resulted in our best scoring official runs [24]. Our initial conclusion is that varying the retrieval model leads to improvement, and especially the combination of different retrieval models hold the promise of making retrieval more effective.

Acknowledgments

We thank Valentin Jijkoun for his help with the Russian collection. Jaap Kamps was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 400-20-036 and 612.066.032. Christof Monz was supported by NWO under project numbers 612-13-001 and 220-80-001. Maarten de Rijke was supported by NWO under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.000.207, and 612.066.032.

References

1. Matthews, P.H.: Morphology. Cambridge University Press (1991)
2. Hollink, V., Kamps, J., Monz, C., de Rijke, M.: Monolingual document retrieval for European languages. Information Retrieval **6** (2003)
3. Harman, D.: How effective is suffixing? Journal of the American Society for Information Science **42** (1991) 7–15
4. Hull, D.: Stemming algorithms – a case study for detailed evaluation. Journal of the American Society for Information Science **47** (1996) 70–84
5. Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In Peters, C., Braschler, M., Gonzalo, J., Kluck, M., eds.: Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001. Volume 2406 of Lecture Notes in Computer Science., Springer (2002) 262–277
6. Kamps, J., Monz, C., de Rijke, M.: Combining evidence for cross-language information retrieval. In Peters, C., Braschler, M., Gonzalo, J., Kluck, M., eds.: Evaluation of Cross-Language Information Retrieval Systems, CLEF 2002. Lecture Notes in Computer Science, Springer (2003)
7. CLEF: Cross language evaluation forum (2003) `http://www.clef-campaign.org/`.
8. Buckley, C., Singhal, A., Mitra, M.: New retrieval approaches using SMART: TREC 4. In Harman, D., ed.: The Fourth Text REtrieval Conference (TREC-4), National Institute for Standards and Technology. NIST Special Publication 500-236 (1996) 25–48

9. Snowball: Stemming algorithms for use in information retrieval (2003) http://www.snowball.tartarus.org/.
10. Frakes, W.: Stemming algorithms. In Frakes, W., Baeza-Yates, R., eds.: Information Retrieval: Data Structures & Algorithms. Prentice Hall (1992) 131–160
11. Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–137
12. Pohlmann, R., Kraaij, W.: Improving the precision of a text retrieval system with compound analysis. In Landsbergen, J., Odijk, J., van Deemter, K., Veldhuijzen van Zanten, G., eds.: Proceedings of the 7th Computational Linguistics in the Netherlands Meeting (CLIN 1996). (1996) 115–129
13. McNamee, P., Mayfield, J.: Character n-gram tokenization for European language text retrieval. Information Retrieval **6** (2003)
14. CLEF-Neuchâtel: CLEF resources at the University of Neuchâtel (2003) http://www.unine.ch/info/clef.
15. Rocchio, Jr., J.: Relevance feedback in information retrieval. In Salton, G., ed.: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs NJ (1971) 313–323
16. Savoy, J.: Combining multiple strategies for effective monolingual and cross-language retrieval. Information Retrieval **6** (2003)
17. Lee, J.: Combining multiple evidence from different properties of weighting schemes. In Fox, E., Ingwersen, P., Fidel, R., eds.: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York NY, USA (1995) 180–188
18. Fox, E., Shaw, J.: Combination of multiple searches. In Harman, D., ed.: The Second Text REtrieval Conference (TREC-2), National Institute for Standards and Technology. NIST Special Publication 500-215 (1994) 243–252
19. Efron, B.: Bootstrap methods: Another look at the jackknife. Annals of Statistics **7** (1979) 1–26
20. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)
21. Worldlingo: Online translator (2003) http://www.worldlingo.com/.
22. PROMT-Reverso: Online translator (2003) http://translation2.paralink.com/.
23. Babylon: Online dictionary (2003) http://www.babylon.com/.
24. Kamps, J., Monz, C., de Rijke, M., Sigurbjörnsson, B.: The University of Amsterdam at CLEF-2003. In Peters, C., ed.: Results of the CLEF 2003 Cross-Language System Evaluation Campaign. (2003) 71–78
25. Schott, H., ed.: Thesaurus Sozialwissenschaften. Informationszentrum Sozialwissenschaften, Bonn (2002) 2 Bände: Alphabetischer und systematischer Teil.
26. Robertson, S., Walker, S., Beaulieu, M.: Experimentation as a way of life: Okapi at TREC. Information Processing & Management **36** (2000) 95–108
27. Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center for Telematics and Information Technology, University of Twente (2001)