Downloaded from UvA-DARE, the institutional repository of the University of Amsterdam (UvA)
http://hdl.handle.net/11245/2.38037

File ID	uvapub:38037
Filename	uams-clef2004-adhoc.pdf
Version	unknown

SOURCE (OR PART OF THE FOLLOWING SOURCE):

Type	book chapter
Title	Effective Translation, Tokenization and Combination for Cross-Lingual Retrieval
Author(s)	J. Kamps, S. Fissaha Adafre, M. de Rijke
Faculty	FNWI: Institute for Logic, Language and Computation (ILLC), FNWI: Informatics Institute (II)
Year	2005

FULL BIBLIOGRAPHIC DETAILS:
http://hdl.handle.net/11245/1.241821

Copyright

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content licence (like Creative Commons).

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)
(pagedate: 2014-11-17)

Effective Translation, Tokenization and Combination for Cross-Lingual Retrieval

Jaap Kamps*, Sisay Fissaha Adafre, and Maarten de Rijke

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: {kamps,sfissaha,mdr}@science.uva.nl

Abstract. Our approach to cross-lingual document retrieval starts from the assumption that effective monolingual retrieval is at the core of any cross-language retrieval system. We devote particular attention to three crucial ingredients of our approach to cross-lingual retrieval. First, effective *tokenization* techniques are essential to cope with morphological variations common in many European languages. Second, effective *combination* methods allow us to combine the best of different strategies. Finally, effective *translation* methods for translating queries or documents turn a monolingual retrieval system into a cross-lingual retrieval system proper. The viability of our approach is shown by a series of experiments in monolingual, bilingual, and multilingual retrieval.

1 Introduction

The CLEF 2004 ad hoc track marked a departure from earlier evaluation campaigns, by its focus on a smaller set of languages, and on lesser known languages [1]. This new set-up prompted us to re-evaluate and extend our earlier approaches to cross-language document retrieval [2, 3, 4]. Our approach to cross-lingual information retrieval starts from the assumption that effective monolingual retrieval is the core of all cross-lingual retrieval tasks [5]. Effective monolingual retrieval requires particular attention to *tokenization*—what document representation is stored in the index? In the context of the CLEF 2004 campaign, we took part in monolingual retrieval for four non-English European languages: Finnish, French, Portuguese, and Russian. Portuguese was new for CLEF 2004. We experimented with a range of language-dependent tokenization techniques, in particular stemming algorithms for all European languages [6], and compound splitting for the compound rich Finnish language. We also experimented with various language-independent tokenization techniques, in particular the use of character n-grams, where we may also index leading and ending character sequences, and retain the original words. Finally, since different document representations have different merits, the use of *combination* methods can be crucial in order to try to get the best of all worlds [7].

* Currently at Archives and Information Studies, Faculty of Humanities, University of Amsterdam.

On top of an effective monolingual retrieval system, one can build a bilingual system by the *translation* of either queries or documents. We performed two in-depth case studies of bilingual retrieval, one for the resource-poor Amharic language, and another for Portuguese. For Portuguese, we performed a comparative analysis of the effectiveness of a number of translation resources. We experimented with machine translation [8] versus a parallel corpus [9], and with query translation versus collection translation. For Amharic we investigated how far we could get by combining the scarcely available resources. Our overall goal in the bilingual experiments was to shed light on the robustness of our monolingual retrieval approaches for various degrees of imperfectly translated queries.

On top of a number of effective bilingual retrieval systems, one can build a multilingual system by *combining* the results in the different languages. We experimented with running English queries on the combined English, Finnish, French, and Russian collections. Here, we experimented with straightforward ways of query translation, using machine translation whenever available, and a translation dictionary otherwise. We also experimented with combination methods using runs made on varying types of indexes.

The rest of this paper is structured as follows. In Section 2 we describe the FlexIR retrieval system used as well as our approaches to tokenization and combination. Section 3 describes our monolingual experiments. Sections 4 (Amharic) and 5 (Portuguese) discuss in detail our bilingual experiments. Section 6 addresses our multilingual experiments. Finally, in Section 7, we offer some conclusions regarding our document retrieval efforts.

2 System Description

All retrieval runs used FlexIR, an information retrieval system developed at the University of Amsterdam. FlexIR supports many types of preprocessing, scoring, indexing, and retrieval models. It also supports several retrieval models, including the standard vector space model, and language models. Our default retrieval model is a vector space model using the Lnu.ltc weighting scheme [10] to compute the similarity between a query and a document. For the experiments on which we report in this paper, we fixed *slope* at 0.2; the pivot was set to the average number of unique words per document. We also experimented with language models [11]. Here, we used a uniform query term importance weight of 0.15.

Blind feedback was applied to expand the original query with related terms. We experimented with different schemes and settings, depending on the various indexing methods and retrieval models used. For our Lnu.ltc runs term weights were recomputed by using the standard Rocchio method [12], 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.

To determine whether the observed differences between two retrieval approaches are statistically significant, we used the bootstrap method, a non-parametric inference test [13, 14]. We take 100,000 samples with replacement

of the topics with their original scores on the two retrieval approaches. We analyze the distribution of improvements over resamples, and look for significant improvements (one-tailed) at significance levels of 0.95 (*); 0.99 (**); and 0.999 (***).

2.1 Tokenization

We carried out extensive experiments with tokenization for monolingual retrieval [5]. These include the following:

Text normalization We do some limited text normalization by removing punctuation, applying case-folding, and mapping diacritics to the unmarked characters. 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 KOI8 (*Kod Obmena Informatsii*), a 1-byte per character encoding. 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 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.

Stop word removal Both topics and documents were stopped using the stopword lists from the Snowball stemming algorithms [6]; for Finnish we used the Neuchâtel stopword list [15]. Additionally, we removed topic specific phrases such as 'Find documents that discuss ...' from the queries. We did not use a "stop stem" or "stop n-gram" list, but we first used a stop *word* list, and then stemmed/n-grammed the topics and documents.

Stemming For all languages we used a stemming algorithm to map word forms to their underlying stems. We used the family of Snowball stemming algorithms, available for all the languages of the CLEF 2004 collections. Snowball is a small string processing language designed for creating stemming algorithms for use in information retrieval [6].

Decompounding For Finnish, a compound-rich language, we apply a decompounding algorithm. We treat all words occurring in the Finnish collection as potential base words for decompounding, and use the associated collection frequencies. We ignore words of length less than 4 as potential compound parts, thus a compound must have at least length 8. As a safeguard against oversplitting, we only consider compound parts with a higher collection frequency than the compound itself. We retain the original compound words, and add their parts to the documents; queries are processed similarly.

n-Gramming For all languages, we used character n-gramming to index all character-sequences of a given length that occur in a word. Unlike stemming, n-gramming is a language-independent approach to morphological normalization. We used three different ways of forming n-grams of length 4. First, we index pure 4-grams. For example, the word `Information` will be indexed as 4-grams `info nfor form orma rmat mati atio tion`. Second, we index 4-grams with leading and ending 3-grams. For the example this will give

`_inf info nfor form orma rmat mati atio tion ion_`. Third, we index 4-grams plus original words. For the example this gives `info nfor form orma rmat mati atio tion information`.

2.2 Run Combination

Combination methods have two distinct purposes. For a number of indexes of the same collection, they can be used to mix evidence from different document representations. For a distributed collection, combination methods can be used to integrate the results for each of the individual subcollections. We combined various 'base' runs using either a weighted or unweighted combination methods. The weighted interpolation was produced as follows. First, we normalized the retrieval status values (RSVs), since different runs may have radically different RSVs. For each run we re-ranked these values in $[0, 1]$ using $RSV_i' = (RSV_i - min_i)/(max_i - min_i)$; this is the Min_Max_Norm considered in [16]. Next, we assigned new weights to the documents using a linear interpolation factor λ representing the relative weight of a run: $RSV_{new} = \lambda \cdot RSV_1 + (1 - \lambda) \cdot RSV_2$. The interpolation factors λ were loosely based on experiments on earlier CLEF data sets [7]. When we combine more than two runs, we give all runs the same relative weight, effectively resulting in the familiar combSUM [17].

3 Monolingual Finnish, French, Portuguese and Russian

In this section we discuss our monolingual retrieval experiments. As explained in the introduction, we view monolingual retrieval as the core of a cross-lingual retrieval system. All other cross-language tasks are performed on top of an a (set of) monolingual indexes. Hence, building an effective monolingual retrieval system is a crucial, first step toward effective bilingual or multilingual retrieval.

3.1 Experiments

All our monolingual runs used the title and description fields of the topics. We constructed five different indexes for each of the languages using *Words*, *Stems*, *4-Grams*, *4-Grams+start/end*, and *4-Grams+Words*:

- *Words*: no morphological normalization is applied, although for Finnish *Split* indicates that words are decompounded.
- *Stems*: topic and document words are stemmed using the morphological tools described in Section 2. For Finnish, *Split+stem* indicates that compounds are split, where we stem the words and compound parts.
- *n-Grams*: both topic and document words are n-grammed, using the settings discussed in Section 2. We have three different indexes: *4-Grams*; *4-Grams+words* where also the words are retained; and *4-Grams+start/end* with beginning and ending 3-grams.

On all these indexes we created runs using the Lnu.ltc retrieval model; on the *Words* and on the *Stems* index we also created runs with a language model, resulting in 7 base runs for French, Portuguese, and Russian.

3.2 Results

Table 1 contains the mean average precision (MAP) scores for all the monolingual 'base' runs described in the previous section. The language model experiments clearly indicate the effectiveness of the stemming algorithm. For the vector space model, there is a small loss for Portuguese, but also a gain in performance for the other three languages. The outcome for the n-gram runs is less clear: there is a substantial gain in effectiveness for Finnish, but no or only a moderate gain for the other three languages. When comparing 4-gram with 4-gram+start/end, we see that including leading and ending 3-grams is always effective. Similarly, including words is effective for three of the four languages.

Table 1. Overview of MAP scores for monolingual base runs. Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	Finnish	*French*	*Portuguese*	*Russian*
Words (baseline)	0.3776	0.4084	0.4032	0.3186
Stems	0.4549*	0.4312*	0.4023	0.3611
4-Grams	0.4949*	0.3673	0.3439	0.2783
4-Grams+start/end	**0.5264*****	0.3794*	0.3653	0.3212
4-Grams+words	0.4930**	0.4133	0.3723	0.3357
Words LM	0.3825	0.4059	0.4040	0.2958
Stems LM	0.4530*	**0.4463**	**0.4269**	**0.3847**

For Finnish we also applied a decompounding algorithm [5], on words and on stems, from which we produced base runs with both the Lnu.ltc retrieval model and a language model, leading to a total of 11 base runs for Finnish. Table 2 contains the MAP scores for the Finnish decompounding experiments. Decompounding leads to improvements for both retrieval models; decompounding and stemming only leads to improvements for the language model run. All Finnish n-gram runs in Table 1 outperform all decompounded runs.

Finally, we experimented with combinations of the base runs just described. For each of the four languages we constructed two combinations of stemmed and n-grammed base runs, as well as a "grand" combination of all base runs. Table 3 lists the MAP scores for our run combinations. For these, the grand combination of all base runs always outperforms the combination of a single (non)stemmed run and a single n-grammed run. When comparing with the best scoring base

Table 2. Overview of MAP scores for Finnish decompounding runs. Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	Words	*Split*	*Stems*	*Split+Stem*
Lnu.ltc	0.3776	0.4329**	0.4549*	0.4414
LM	0.3825	0.4021	0.4530*	**0.4617***

Table 3. Overview of MAP scores for our run combinations. Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	Finnish	*French*	*Portuguese*	*Russian*
4-Grams+words;(Split+)stem	$0.4787^{\star\star}$	0.4410	0.4110	$0.4227^{\star\star}$
4-Grams+start/end;(Split+)words	$0.5007^{\star\star\star}$	0.4092	0.4180	$0.4058^{\star\star}$
All base runs	$\mathbf{0.5203}^{\star\star\star}$	$\mathbf{0.4499}^{\star}$	**0.4326**	$\mathbf{0.4412}^{\star\star}$

runs in Tables 1, we see that there is only a substantial improvement for Russian. There is a moderate improvement for French and Portuguese. The best Finnish n-gram run even outperforms the grand combination.

4 Bilingual Retrieval: Amharic to English

In Amharic, which belongs to the Semitic family of languages, word formation involves affixation, reduplication, Semitic stem interdigitation, among others. The most characteristic feature of Amharic morphology is root-pattern phenomena. This is especially true of Amharic verbs, which rely heavily on the arrangement of consonants and vowels in order to code different morphosyntactic properties (such as perfect, imperfect, etc.). Consonants, which mostly carry the semantic core of the word, form the root of the verb. Consonants and vowel patterns together constitute the stems, and stems take different types of affixes (prefixes and suffixes) to form the fully inflected words; see [18].

For our bilingual Amharic to English runs, we attempted to show how the scarce resources for Amharic can be used in (Amharic-English) bilingual information retrieval settings. Since English is used on the document side, it is interesting to see how the existing retrieval techniques can be optimized in order to make the best use of the output of the error-prone translation component.

Our Amharic to English query translation is based mainly on dictionary look up. We used an Amharic-English bilingual dictionary which consists of 15,000 fully inflected words. Due to the morphological complexity of the language, we expected the dictionary to have limited coverage. In order to improve on the coverage, two further dictionaries, root-based and stem-based, were derived from the original dictionary. We also tried to augment the dictionary with a bilingual lexicon extracted from aligned Amharic-English Bible text. However, most of the words are old English words and are also found in the dictionary. The word dictionary also contains commonly used Amharic collocations. Multiword collocations were identified and marked in the topics. For this purpose, we used a list of multiword collocations extracted from an Amharic text corpus. The dictionaries were searched for a translation of Amharic words in the following order: word-dictionary, stem dictionary, root dictionary.

Leaving aside the ungrammaticality of the output of the above translation, there are a number of problems. One is the problem of unknown words. The words may be Amharic words not included in the dictionary or foreign words. Some foreign words and their transliteration have the same spelling or are nearly

Table 4. Coverage of the respective techniques over the words occurring in the Amharic topics.

Total no. of words	Word dictionary	Root dictionary	English spell checker
1,893	813	178	57

identical. To take advantage of this fact, Amharic words not in the dictionary are checked using an English spellchecker (Aspell). We process the English words suggested by the spellchecker one by one, and if the suggestion is similar enough to the Amharic word, it will be taken as a translation. Specifically, we look for similarity in length (i.e., a difference in length < 4), and for string similarity (i.e., a longest common substring ration of > 0.7). In this way, we address the typographical variations between the English word and its transliteration. Other unknown words are simply passed over to the English translation. Another problem relates to the selection of the appropriate translation from among the possible translations found in the dictionary. In the absence of frequency information, the most frequently used English word is selected as a translation of the corresponding Amharic word. This is achieved by querying the web. The coverage of the translation is 55%. The number of correct translations is still lower. Table 4 gives some idea of the coverage of the translation strategy.

4.1 Experiments

In our experiments we focus on the translation of the Amharic topics to English as detailed above. We used a similar set of indexes as for the monolingual runs described earlier (*Words*, *Stems*, *4-Grams*, *4-Grams+start/end*, *4-Grams+words*). For all of these, Lnu.ltc runs were produced, and for the *Word* and *Stems* indexes we also produced a language model run, leading to 7 base runs for the Amharic to English task. Additionally, we created two run combinations: a combination of the stemmed and an n-grammed run, and a combination of all base runs.

4.2 Results

Table 5 shows the mean average precision scores for our base runs. For the resource-poor Amharic to English task, we expected a fairly low performance, somewhere in the 0.12–0.20 range. However, the vector space model run on the *Words* index is surprisingly effective. Furthermore, n-gramming leads to a significant loss of performance. Table 5 also lists results on run combinations. The combination of a stemmed and a n-grammed run does not lead to improvement. The combination of all base runs leads to the best performance for Amharic to English, but the score is not significantly better than for the word-based run.

5 Bilingual Retrieval: English to Portuguese

Having discussed experiments with the resource-poor language of Amharic in the previous section, we now focus on bilingual retrieval for Portuguese. We

Table 5. Overview of MAP scores for Amharic to English runs. Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	Amharic to English
Words (baseline)	0.2071
Stems	0.1961
4-Grams	$0.1224^{\star\star}$
4-Grams+start/end	$0.1300^{\star}$
4-Grams+words	$0.1467^{\star\star\star}$
Words LM	0.1694
Stems LM	0.1703
4-Grams+words;Stems	0.1915
All base runs	**0.2138**

evaluate the relative effectiveness of various translation methods for English to Portuguese retrieval. All our runs used the title and description fields of the topics. For our bilingual runs, we experimented with the WorldLingo machine translation [8] for translations into Portuguese, with a parallel corpus for translations into Portuguese.

Machine Translation. We used the WorldLingo machine translation [8] for translating the English topics into Portuguese. The translation is actually in Brazilian Portuguese, but for retrieval purposes the linguistic differences between Portuguese and Brazilian are fairly limited.

Parallel Corpus. We used the sentence-aligned parallel corpus [9], based on the Official Journal of the European Union [19]. We built a Portuguese to English translation dictionary, based on a word alignment in the parallel corpus. Since the word order in English and Portuguese are not very different, we only considered potential alignments with words in the same position, or one or two positions off. We ranked potential translations with a score based on:

- *Cognate matching* Reward similarity in word forms, by looking at the number of leading characters that agree in both languages.
- *Length matching* Reward similarity in word lengths in both languages.
- *Frequency matching* Reward similarity in word frequency in both languages.

To further aid the alignment, we constructed a list of 100 most frequent Portuguese words in the corpus, and manually translated these to English. The alignments of these highly frequent words were resolved before the word alignment phase. We built a Portuguese to English translation dictionary by choosing the most likely translation, where we only include words that score above a threshold. The length of the translation dictionary is 19,554 words. We use the translation dictionary resulting from the parallel corpus for two different purposes. Firstly, we translate the English topics into Portuguese. Secondly, we translate the Portuguese collection into English.

Table 6. Overview of MAP scores for all English to Portuguese runs. Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	query EU	*query Wordlingo*	*collection EU*
Words (baseline)	0.2641	0.3220	0.3830
Stems	0.3201**	0.3281	0.3901
4-Grams	0.2134	0.2856	0.3704
4-Grams+start/end	0.2296	0.2856	0.3826
4-Grams+words	0.2355	0.3203	0.3678
Words LM	0.2511	0.3167	0.3471
Stems LM	0.2993	0.3257	0.3835
4-Grams+words;Stems	0.2755	0.3207	0.3850
All base runs		**0.4366**	

5.1 Experiments

For Portuguese we used a similar set of indexes as described earlier (*Words*, *Stems*, *4-Grams*, *4-Grams+start/end*, *4-Grams+words*). We produce runs with the Lnu.ltc retrieval model, and for the *Word* and *Stems* indexes we also produced a language model run. Additionally, for the English to Portuguese task we used three types of translation: query translation using machine translation (WorldLingo), query translation using a parallel corpus (query EU), and collection translation using a parallel corpus (collection EU). This gave rise to a total of 21 base runs for the English to Portuguese task. Finally, for each of the three translation methods, we look at the combination of the stemmed and a n-grammed run, and we also look at the combination of all 21 base runs.

5.2 Results

Table 6 shows the mean average precision scores for our base runs. Comparing the different translation methods for the plain *Words* index, we see that, for query translation, the machine translation is more effective than the parallel corpus. This is no surprise, since a word by word translation dictionary was derived from the parallel corpus. However, if the parallel corpus is used to translate the collection, we obtain a higher score for the *Words* index than both query translation methods. Applying a stemming algorithm is helpful for the MAP score for all three ways of translation. The use of n-gramming is not effective for any of the translation methods. Table 6 also lists results for run combinations. The combination of a stemmed and a n-grammed run only leads to improvement for the collection translation method. The combination of all base runs leads to the best performance for English to Portuguese. The resulting score for English to Portuguese is impressive, outperforming our best monolingual score.

6 Multilingual Retrieval

Based on the experience of monolingual and bilingual experiments discussed above, we now turn to the "grand" task in the ad hoc track: multilingual re-

trieval. In CLEF 2004, the target collection was the combined English, Finnish, French, and Russian collections. We experimented with a fairly straightforward approach to query translation, using machine translation if available and otherwise resorting to a translation dictionary.

6.1 Experiments

We submitted a total of 4 multilingual runs, all using the title and description fields of the English topic set. The multilingual runs were based on the following mono- and bilingual runs:

- *English to English* – This is just a monolingual run, similarly processed as the other monolingual runs discussed above.
- *English to Finnish* — We translated the English topics into Finnish using the Mediascape on-line dictionary [20]. For words present in the dictionary, we included all possible translations available. For words not present in the dictionary, we simply retained the original English words.
- *English to French* — We translated the English topics into French using the WorldLingo machine translation [8].
- *English to Russian* — Again, we translated the English topics into Russian using the WorldLingo machine translation [8].

We applied a straightforward combination method to the results of the mono- and bilingual runs just described. We use an unweighted combSUM of the following sets of runs: The single *4-Grams+words* run for each of the four languages; both a *Words LM* and a *Stems LM* run for each of the four languages; three runs (*Words*, *Stems*, and *4-Grams+start/end*) for each of the four languages; all seven runs for each of the four languages.

6.2 Results

Table 7 shows our mean average precision scores for all base runs used in the multilingual task. We did not apply decompounding to the Finnish topics. As an aside, we see that for monolingual English, the language model is particularly effective. The results for Finnish, French, and Russian are generally in line with the monolingual results discussed above, be it that the n-gramming approaches are generally more effective on the translated topics. Table 7 also includes the run combinations that result in the multilingual runs. Recall that all these combinations are unweighted. On the whole, the performance increases with the number of runs included in the combination.

7 Discussion and Conclusions

In this paper, we reported on a range of cross-lingual retrieval experiments. Our approach is rooted on building effective monolingual retrieval systems. We

Table 7. Overview of MAP scores for all multilingual runs (bottom half) and of the mono- and bilingual runs used to produce them (top half). Best scores are in boldface, stars indicate a significant improvement over the word-based run.

	English	*Finnish*	*French*	*Russian*
Words (baseline)	0.4488	0.2057	0.3351	0.2012
Stems	0.4885*	**0.2719***	0.3677*	0.1478
4-Grams	0.3986*	0.2376	0.3585	0.2140
4-Grams+start/end	0.4369	0.2578	**0.3810**	**0.2623**
4-Grams+words	0.4387	0.2270	0.3596	0.2595
Words LM	0.4909	0.1913	0.3489	0.1935
Stems LM	**0.5156***	0.2303	0.3676	0.1978
4-Grams+words	0.2333			
Words LM;Stems LM	0.3040			
Words;Stems;4-Grams+start/end	0.3258			
All	**0.3427**			

performed a comparative analysis of a range of tokenization techniques for monolingual retrieval in Finnish, French, Portuguese, and Russian, shedding light on the relative effectiveness of each of the methods. Since different document representations have different merits, combination methods can be extremely useful to combine the different sources of evidence.

With the translation of either queries or documents, a create a bilingual retrieval system. We investigated the robustness of our approach by focusing the resource-poor Amharic language. Making use of the scarcely available resources results in an error-prone translation. Much to our surprise, the retrieval results are fair. We also investigated one of the world's major languages, Portuguese, and examined the relative effectiveness of different translation resources, and of query versus collection translation. Our results indicate interesting differences between the bilingual approaches. The effectiveness of combining different translation methods was highlighted by the fact that the best bilingual score outperformed the best monolingual score.

Combination methods are also crucial in retrieving from a distributed multilingual collection. For multilingual retrieval from the combined English, Finnish, French, and Russian collections, we experimented with straightforward query translations for the translation of the English queries into Finnish, French, and Russian. Using only straightforward unweighted run combination methods, we constructed multilingual runs. Our results indicate that including a range of different document representations per language is generally beneficial.

Acknowledgments We want to thank Valentin Jijkoun, Gustavo Lacerda de Melo, and Willem Robert van Hage. Sisay Fissaha Adafre was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Jaap Kamps was supported by a grant from NWO under project number 612.066.302. Maarten de Rijke was supported by grants from NWO, under

project numbers 365-20-005, 612.069.006, 612.000.106, 220-80-001, 612.000.207, 612.066.302, 264-70-050, and 017.001.190.

References

[1] Peters, C., Braschler, M., Di Nunzio, G., Ferro, N.: CLEF 2004: Ad hoc track overview and results analysis. In: Fifth Workshop of the Cross-Language Evaluation Forum (CLEF 2004), Springer (2005)

[2] Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In: Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001, Springer (2002) 262–277

[3] Kamps, J., Monz, C., de Rijke, M.: Combining evidence for cross-language information retrieval. In: Evaluation of Cross-Language Information Retrieval Systems, CLEF 2002, Springer (2003) 111–126

[4] Kamps, J., Monz, C., de Rijke, M., Sigurbjörnsson, B.: Language-dependent and language-independent approaches to cross-lingual text retrieval. In: Comparative Evaluation of Multilingual Information Access Systems, CLEF 2003, Springer (2004)

[5] Hollink, V., Kamps, J., Monz, C., de Rijke, M.: Monolingual document retrieval for European languages. Information Retrieval **7** (2004) 33–52

[6] Snowball: Stemming algorithms for use in information retrieval (2004) `http://www.snowball.tartarus.org/`.

[7] Kamps, J., de Rijke, M.: The effectiveness of combining information retrieval strategies for European languages. In: Proceedings of the 2004 ACM Symposium on Applied Computing, ACM Press (2004) 1073–1077

[8] Worldlingo: Online translator (2004) `http://www.worldlingo.com/`.

[9] Koehn, P.: European parliament proceedings parallel corpus 1996-2003 (2004) `http://people.csail.mit.edu/people/koehn/publications/europarl/`.

[10] Buckley, C., Singhal, A., Mitra, M.: New retrieval approaches using SMART: TREC 4. In: The Fourth Text REtrieval Conference (TREC-4), National Institute for Standards and Technology. NIST Special Publication 500-236 (1996) 25–48

[11] Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center for Telematics and Information Technology, University of Twente (2001)

[12] Rocchio, Jr., J.: Relevance feedback in information retrieval. In: The SMART Retrieval System. Prentice-Hall, Englewood Cliffs NJ (1971) 313–323

[13] Efron, B.: Bootstrap methods: Another look at the jackknife. Annals of Statistics **7** (1979) 1–26

[14] Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)

[15] CLEF-Neuchâtel: CLEF resources at the University of Neuchâtel (2004) `http://www.unine.ch/info/clef`.

[16] Lee, J.: Combining multiple evidence from different properties of weighting schemes. In: 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

[17] Fox, E., Shaw, J.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2), National Institute for Standards and Technology. NIST Special Publication 500-215 (1994) 243–252

[18] Nega, A.: Development of Stemming Algorithm for Amharic Text Retrieval. PhD thesis, University of Sheffield (1999)
[19] European Union: Official Journal of the European Union (2004) http://europa.eu.int/eur-lex/.
[20] Mediascape: English-Finnish-English on-line dictionary (2004) http://efe.scape.net/.