Published in Lecture Notes in Computer Science 3237, 322-336, 2004
which should be used for any reference to this work

Report on CLEF-2003 Monolingual Tracks: Fusion of Probabilistic Models for Effective Monolingual Retrieval

Jacques Savoy

Institut interfacultaire d'informatique, Université de Neuchâtel,
Pierre-à-Mazel 7, 2001 Neuchâtel, Switzerland
Jacques.Savoy@unine.ch
http://www.unine.ch/info/clef/

Abstract. For our third participation in the CLEF evaluation campaign, our first objective was to propose more effective and general stopword lists for the Swedish, Finnish and Russian languages, along with an improved, more efficient and simpler stemming procedure for these three languages. Our second goal was to suggest a combined search approach based on a data fusion strategy that would work with various European languages. Included in this combined approach is a decompounding strategy for the German, Dutch, Swedish and Finnish languages.

1 Introduction

Based on our experiments of the previous year [11], in CLEF 2003 we participated in the French, Spanish, German, Italian, Dutch, Swedish, Finnish and Russian monolingual tasks without relying on a dictionary. This paper presents the approaches we used in the monolingual track and is organized as follows. Section 2 contains an overview of the nine test collections used while Section 3 describes our general approach to building stopword lists and stemmers for use with languages other than English. In Section 4, we suggest a simple decompounding algorithm that can be used for German, Dutch, Swedish and Finnish. Section 5 evaluates two probabilistic models and nine vector-space schemes using the nine test collections. Finally, Section 6 evaluates various data fusion operators, and presents our official runs.

2 Overview of the Test Collections

The corpora used in our experiments included newspapers such as the *Los Angeles Times* (1994, English), *Glasgow Herald* (1995, English), *Le Monde* (1994, French), *La Stampa* (1994, Italian), *Der Spiegel* (1994/95, German), *Frankfurter Rundschau* (1994, German), *NRC Handelsbald* (1994/95, Dutch), *Algemeen Dagblad* (1995/95, Dutch), *Tidningarnas Telegrambyrå* (1994/95, Swedish), *Aamulehti* (1994/95, Finnish), and *Izvestia* (1995, Russian). Additional sources of

information consisted of news agency documents such as *EFE* (1994/95, Spanish) and the Swiss news agency (1994/95, available in French, German and Italian but without parallel translation).

As shown in Tables 1 and 2, these corpora are of various sizes, with the Spanish collection being the biggest and the German, English and Dutch collections next in size. Ranking third are the French, Italian and Swedish corpora, then somewhat smaller is the Finnish collection and finally the Russian collection is clearly the smallest. Across all the corpora the mean number of distinct indexing terms per document is relatively similar (around 112), although this number is slightly larger for the English collection (156.9) and smaller for the Swedish corpus (79.25).

Table 1. Test collection statistics

	English	French	German	Spanish
Size (in MB)	579 MB	331 MB	668 MB	1,086 MB
# of documents	169,477	129,806	294,809	454,045
# of distinct terms	426,757	355,691	1,666,538	774,263
Number of distinct indexing terms / document				
Mean	156.9	118.5	111.9	112.9
Standard deviation	118.77	95.72	100.06	55.75
Median	129	89	84	100
Maximum	1,881	1,621	2,424	642
Minimum	2	3	1	5
Number of queries	54	52	56	57
Number of rel. items	1,006	946	1,825	2,368
Mean rel. items / request	18.63	18.19	32.59	41.54
Standard deviation	28.61	33.16	36.95	57.37
Median	7	8	24	22
Maximum	139	193	226	303
Minimum	1	1	1	1

Tables 1 and 2 also compare the number of relevant documents per request, with the mean always being greater than the median (e.g., for the English collection, the average number of relevant documents per query is 18.63 with the corresponding median being 7). These findings indicate that each collection contains numerous documents, yet only a rather small number of relevant items are found per query. For each collection, 60 queries were created. However, relevant documents are not found for each request and each language. For the English collection, Queries #149, #161, #166, #186, #191, and #195 do not have any relevant items; for the French corpus, requests with no relevant documents are #146, #160, #161, #166, #169, #172, #191, #194; for the German collection: Queries #144, #146, #170, #191; for the Spanish collection: Queries #169, #188, #195; for the Italian collection: Queries #144, #146, #158, #160, #169, #170, #172, #175, #191; for the Dutch collection: Queries #160, #166, #191, #194; for the Swedish collection: Queries #146,

Table 2. Test collection statistics

	Italian	Dutch	Swedish	Finnish	Russian
Size (in MB)	363 MB	540 MB	352 MB	137 MB	68 MB
# of documents	157,558	190,604	142,819	55,344	16,716
# of distinct terms	560,087	883,953	767,504	1,444,232	345,728
Number of distinct indexing terms / document					
Mean	116.4	110	79.25	114	124.5
Standard deviation	88.24	107.03	64.00	91.35	124.53
Median	84	77	62	87	41
Maximum	1,395	2,297	1,547	1,946	1,769
Minimum	1	1	1	1	1
Number of queries	51	56	54	45	28
Number of rel. items	809	1,577	1,006	483	151
Mean rel. items / request	15.86	28.16	18.63	10.73	5.39
Standard deviation	20.32	43.10	28.35	15.78	7.11
Median	8	14.5	11.5	5	3
Maximum	110	226	170	82	31
Minimum	1	1	1	1	1

#160, #167, #191, #194, #198; for the Finnish corpus: Queries #141, #144, #145, #146, #160, #167, #169, #175, #182, #186, #188, #189, #191, #194, #195. The Russian corpus appeared for the first time in a CLEF evaluation campaign and only 28 requests actually found relevant documents.

During the indexing process of our automatic runs, we retained only the following logical sections from the original documents: <TITLE>, <HEADLINE>, <TEXT>, <LEAD>, <LEAD1>, <TX>, <LD>, <TI>, and <ST>. From the topic descriptions we automatically removed certain phrases such as "Relevant document report ...", "Find documents that give ...", "Trouver des documents qui parlent ...", "Sono valide le discussioni e le decisioni ...", "Relevante Dokumente berichten ..." or "Los documentos relevantes proporcionan información ...".

3 Stopword Lists and Stemming Procedures

In order to define general stopword lists, we first accounted for the top 200 most frequent words found in the various languages, together with articles, pronouns, prepositions, conjunctions or very frequently occurring verb forms (e.g., to be, is, has, etc.). With respect to the stopword lists we used last year [11], we only modified those for Swedish and Finnish, and created a new list for Russian (these lists are available at `www.unine.ch/info/clef/`). For English we used the list provided by the SMART system (571 words), while for the other European languages, our stopword list contained 430 words for Italian, 463 for French, 603 for German, 351 for Spanish, 1,315 for Dutch, 747 for Finnish, 386 for Swedish and 420 for Russian.

Once it removes high-frequency words, an indexing procedure generally applies a stemming algorithm in an attempt to conflate word variants into the same stem or root. In developing this procedure for various European languages, we first removed only inflectional suffixes such as singular and plural word forms, and also feminine and masculine forms, so that they conflate to the same root. Our stemmers also try to reduce various word declensions to the same stem, such as those used in the German, Finnish and Russian languages.

More sophisticated schemes have already been proposed for the removal of derivational suffixes (e.g., "-ize", "-ably", "-ship" in the English language), as can be seen in the stemmer developed by Lovins [8] (based on a list of over 260 suffixes), or that of Porter [9] (which looks for about 60 suffixes). For the French language only, our stemming approach tried to remove some derivational suffixes (e.g., "communicateur" → "communiquer", "faiblesse" → "faible"). For the Dutch language we used Kraaij & Pohlmann's stemmer [7]. Our various stemming procedures can be found at `www.unine.ch/info/clef/`. Currently, it is not clear whether a stemming procedure such as ours that removes only inflectional suffixes from nouns and adjectives is sufficient, or whether better retrieval effectiveness may be achieved by a stemming approach that also accounts for verbs or that removes both inflectional and derivational suffixes.

Finally diacritic characters, not usually not present in English collections (with some exceptions, such as " résumé"), but very common in Italian, Dutch, Finnish, Swedish, German, Spanish and Russian, were replaced by their corresponding non-accentuated letter. For this last language, we converted and normalized the Cyrillic Unicode characters into the Latin alphabet (the Perl script is available at `www.unine.ch/info/clef/`).

4 Decompounding Words

Most European languages manifest other morphological characteristics in addition to inflection, with compound word constructions being just one example (e.g., handgun, worldwide). In German, for example, compound words are widely used and can cause more difficulties than in English. For example, an insurance company would be "Versicherungsgesellschaft" ("Versicherung" + "s" + "Gesellschaft"). However the morphological marker ("s") is not always present (e.g., "Atomtests" built as "Atom" + "Tests"), and sometimes the letter "S" belongs to the decompounded word (e.g., "Wintersports" for "Winter" + "Sports"). In Finnish, we also encounter similar constructions as such as "rakkauskirje" ("rakkaus" + "kirje" for love & letter) or "työviikko" ("työ" + "viikko" for work & week). Recently, Braschler [3] showed that decompounding German words may significantly improve retrieval performance.

Our proposed decompounding approach shares some similarity with Chen's algorithm [5]. Before using it, we create a word list composed of all words appearing in the given collection (without stemming). Associated with each word, we also store the number of its occurrences in the collection (some examples are given in Table 3).

Table 3. Examples of German words included in our word list

computer	2,452	port	1,091
computers	79	ports	2
sicherheit	6,583	sport	1,483
sicher	4,522	sports	199
heit	4		
		winter	1,643
bank	9,657	winters	148
bund	7,032	wintersport	44
bundes	2,884	wintersports	2
bundesbank	1,453		
präsident	24,041		

In order to present an overview of our decompounding approach, we will take as an example the German word "Computersicherheit," composed of "Computer" + "Sicherheit" (security). This compound word does not appear in our German word list as shown in Table 3, so our algorithm starts the decompounding process by attempting to split a word following the $k = 4$ last letters (given the two strings "computersicher" and "heit"). During the entire procedure, we only consider words having a length greater than a given threshold (fixed at 3 for all languages in our experiments). If both components appear in the word list, then we have a candidate for decompounding; otherwise the k limit is increased by one. Since, in our case, the string "computersiche" does not appear in the German word list, splitting is rejected. When $k = 9$, our algorithm will find the word "computers" in the word list, but will fail to find the word "icherheit". With $k = 10$, our algorithm will find both the word "computer" and "sicherheit" in the German word list (see Table 3) and this solution becomes the top level decompounding suggestion. Recursively, the system now tries to decompose the two parts, namely the words "computer" and "sicherheit". During this recursive process, the system is allowed to ignore some short sequences of letters at the end of a word (such as "-s" or "-es" in German, or "-s" for the Swedish language) because such morphological markers may indicate the genitive form (such as "'s" in the noun phrase "John's book").

After this generative part, the system responds with a tree of possible ways in which the compound construction can be broken down and for each component, we find the number of its occurrences in the corpus. In our example, the answer will be (computer 2452, sicherheit 6583 (sicher 4522, heit 4)). Thus, from this result, we know that the word "Sicherheit" appears 6,583 times in the corpus, and we can consider decomposing this term into the words "sicher" and "heit". From this we can add (or replace) the compound word in the document (or in the request) by all possible candidates ("computer" + "sicherheit", and "computer" + "sicher" + "heit" in our case) or by decompounding only the minimum number of terms ("computer" + "sicherheit" in our case).

However, when faced with multiple candidates, our algorithm will try to select the single "best" one. To achieve this, our system considers the total number

of occurrences for the component words and, if this value is greater than the number of occurrences for the compound construction, the candidate will be selected. In our example, the system will not decompound the word "Sicherheit" because the number of occurrences of the words "sicher" (4,522) and "heit" (4) will not produce a total (4,526) greater than the number of occurrences of the word "sicherheit" (6,583).

If we consider the German word "Bundesbankpräsident" (president of the (German) federal bank), the generative part of our algorithm would return (bundesbank 1453 (bund 7032, bank 9657), präsident 24041) and the final decompounding approach would return (bund 7032, bank 9657, präsident 24041). In this case, the number of occurrences of "bundesbank" (1,453) is smaller than the sum of the occurrences of the words "bund" and "bank". However, our approach does not always generate the appropriate components of a compound term. For example, faced with the compound construction "wintersports", the system answers with (winter 1643, port 1091) instead of (winter 1643, sport 1483). This problem is due to the fact that the first part of our approach ignores backtracking and will stop when it encounters the first splitting of the compound into two parts.

5 Indexing and Searching Strategy

In order to obtain a broader view of the relative merits of various retrieval models, we first adopted a binary indexing scheme by which each document (or request) is represented by a set of keywords, without any weight. To measure the similarity between documents and requests, we computed the inner product (retrieval model denoted "doc=bnn, query=bnn" or "bnn-bnn"). In order to weight the presence of each indexing term in a document surrogate (or in a query), we can compute the term occurrence frequency (retrieval model notation: "doc=nnn, query=nnn" or "nnn-nnn") or we can compute the term frequency in the collection (or more precisely the inverse document frequency, denoted by idf_j). Cosine normalization can prove beneficial and each indexing weight can vary within the range of 0 to 1 (retrieval model notation: "ntc-ntc", Table 4 shows the exact weighting formulation).

Other variants might also be created. For example, the tf component may be computed as $0.5 + 0.5 \cdot$ [tf / max tf in a document] (retrieval model denoted "doc=atn"). We might also consider whether a term's presence in a shorter document provides stronger evidence than it does in a longer document, leading to more complex IR models; for example, the IR model denoted by "doc=Lnu" [4], "doc=dtu"[12]. In addition to the previous models based on the vector-space approach, we also considered probabilistic models. In this respect, we used the Okapi probabilistic model [10]. In Table 4, n denotes the number of documents in the collection, nt_i indicates the number of distinct indexing terms included in the representation of D_i, l_i the length of D_i measured as the sum of tf_{ij}, and $avdl$ the mean document length.

Table 4. Weighting schemes

bnn	$w_{ij} = 1$	npn	$w_{ij} = tf_{ij} \cdot ln\left[\frac{n - df_j}{df_j}\right]$
nnn	$w_{ij} = tf_{ij}$		
ntc	$w_{ij} = \frac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t}(tf_{ik} \cdot idf_k)^2}}$	atn	$w_{ij} = idf_j \cdot \left[\frac{0.5 + 0.5 \cdot tf_{ij}}{max\ tf_{i.}}\right]$
lnc	$w_{ij} = \frac{ln(tf_{ij})+1}{\sqrt{\sum_{k=1}^{t}(ln(tf_{ik})+1)^2}}$	dtn	$w_{ij} = (ln(ln(tf_{ij}) + 1) + 1) \cdot idf_j$
ltn	$w_{ij} = (ln(tf_{ij}) + 1) \cdot idf_j$	ltc	$w_{ij} = \frac{(ln(tf_{ij})+1) \cdot idf_j}{\sqrt{\sum_{k=1}^{t}[(ln(tf_{ik})+1) \cdot idf_k]^2}}$
Okapi	$w_{ij} = \frac{(k_1+1) \cdot tf_{ij}}{K + tf_{ij}}$ with $K = k_1 \cdot \left[(1-b) + b \cdot \frac{l_i}{avdl}\right]$		
dtu	$w_{ij} = \frac{(ln(ln(tf_{ij})+1)+1) \cdot idf_j}{(1-slope) \cdot pivot + (slope \cdot nt_i)}$		
Lnu	$w_{ij} = \frac{\frac{ln(tf_{ij})+1}{ln\left(\frac{l_i}{nt_i}\right)+1}}{(1-slope) \cdot pivot + (slope \cdot nt_i)}$		

As a second probabilistic approach, we implemented the Prosit (PRObabilistic Sift of Information Terms) approach [1], [2] which is based on the following indexing formula:

$$
\begin{aligned}
w_{ij} &= Inf^1_{ij} \cdot Inf^2_{ij} = (1 - Prob^1_{ij}) \cdot Inf^2_{ij} \text{ with} \\
Prob^1_{ij} &= tfn_{ij} \ / \ (tfn_{ij} + 1) \\
tfn_{ij} &= tf_{ij} \cdot \log_2 [1 + ((C \cdot mean\ dl) \ / \ l_j)] \\
Inf^2_{ij} &= -\log_2 [1/(1 + l_j)] - tfn_{ij} \cdot \log_2 [l_j/(1 + l_j)] \text{ with } l_j = tc_j/n
\end{aligned}
$$

where tc_j indicates the number of occurrences of term t_j in the collection and n the number of documents in the corpus. In our experiments, the constants b, k_1, $avdl$, C and $mean\ dl$ are fixed according to values listed in Table 5 while the constant $pivot$ is fixed at 100, and $slope$ at 0.1.

To evaluate our approaches, we used the SMART system as a test-bed running on an Intel Pentium III/600 (memory: 1 GB, swap: 2 GB, disk: 6 x 35 GB). To measure the retrieval performance, we adopted the non-interpolated mean average precision (computed on the basis of 1,000 retrieved items per request by the TREC-EVAL program, see ftp://ftp.cs.cornell.edu/pub/smart/). We indexed the English, French, Spanish and Italian collections using words as indexing units. The evaluation of our two probabilistic models and nine vector-space schemes is given in Table 6.

In order to represent German, Dutch, Swedish, Finnish and Russian documents and queries, we considered the n-gram, decompounding and word-based indexing schemes. The resulting mean average precision for these various indexing approaches is shown in Table 7 (German and Dutch corpora), in Table 8 (Swedish and Finnish languages) and in Table 9 (Russian collection).

Table 5. Parameter setting for the various test collections

Language	Index	b	k_1	$avdl$	C	$mean\ dl$
English	word	0.8	2	800	1.5	167
French	word	0.75	3	900	1.25	182
Spanish	word	0.4	1.2	400	1.75	157
German	word	0.5	1.5	600	3	152
German	5-gram	0.3	1	500	2.5	475
Italian	word	0.55	1.5	800	1.25	165
Dutch	word	0.8	3	600	2.25	110
Dutch	5-gram	0.6	1.2	600	1.75	362
Finnish	word	0.75	2	900	1.25	114
Finnish	5-gram	0.6	1.2	800	2	539
Swedish	word	0.7	2	500	3	79
Swedish	4-gram	0.75	2	900	1.75	292
Russian	word	0.7	2	800	1.5	124
Russian	5-gram	0.75	1.2	750	1.75	451
Russian	4-gram	0.75	1.2	750	1.75	468

Table 6. Mean average precision of various single searching strategies (monolingual)

	Mean average precision			
Query TD Model	English 54 queries	French 52 queries	Spanish 57 queries	Italian 51 queries
Prosit	48.19	**52.01**	47.23	47.17
doc=Okapi, query=npn	**48.83**	51.64	**48.85**	**48.80**
doc=Lnu, query=ltc	44.51	48.26	45.79	45.32
doc=dtu, query=dtn	43.17	46.58	45.03	45.71
doc=atn, query=ntc	45.55	45.48	44.04	45.77
doc=ltn, query=ntc	34.68	39.01	42.40	42.56
doc=ntc, query=ntc	27.12	32.74	27.08	28.90
doc=ltc, query=ltc	28.14	34.41	29.74	28.63
doc=lnc, query=ltc	33.89	37.98	33.52	32.68
doc=bnn, query=bnn	15.97	24.01	26.48	25.33
doc=nnn, query=nnn	6.50	12.27	19.84	22.36

It was observed that pseudo-relevance feedback (blind-query expansion) seems to be a useful technique for enhancing retrieval effectiveness. In this study, we adopted Rocchio's approach [4] with $\alpha = 0.75$, $\beta = 0.75$ whereby the system was allowed to add m terms extracted from the k best ranked documents from the original query. To evaluate this proposition, we used the Okapi and the Prosit probabilistic models and enlarged the query by the 10 to 175 terms provided by the 3 or 10 best-retrieved articles.

The results shown in Tables 10, 11, 12, and 13 (giving our best results) indicate that the optimal parameter setting seems to be collection-dependent. Moreover, performance improvement also seems to be collection dependent (or

Table 7. Mean average precision of various single searching strategies (German & Dutch collections)

	Mean average precision					
Query TD Model	German words 56 queries	German decomp. 56 queries	German 5-gram 56 queries	Dutch words 56 queries	Dutch decomp. 56 queries	Dutch 5-gram 56 queries
Prosit	42.14	45.53	42.88	**47.15**	48.36	39.41
Okapi-npn	**44.54**	**46.93**	**44.27**	46.86	**48.73**	**40.23**
Lnu-ltc	40.64	45.44	39.63	43.38	45.08	33.63
dtu-dtn	42.60	43.95	39.08	42.69	43.78	33.82
atn-ntc	40.98	43.67	40.36	41.92	43.52	36.43
ltn-ntc	39.07	39.32	38.57	38.45	39.51	32.47
ntc-ntc	27.40	32.64	31.59	29.27	30.36	29.42
ltc-ltc	28.85	36.02	32.76	30.97	32.41	28.24
lnc-ltc	30.16	35.93	32.10	31.39	33.15	28.53
bnn-bnn	23.63	23.31	21.07	26.14	26.80	21.16
nnn-nnn	15.97	10.85	9.78	11.35	10.64	9.82

Table 8. Mean average precision of various single searching strategies (Swedish & Finnish collections)

	Mean average precision					
Query TD Model	Swedish words 54 queries	Swedish decomp. 54 queries	Swedish 4-gram 54 queries	Finnish words 45 queries	Finnish decomp. 45 queries	Finnish 5-gram 45 queries
Prosit	39.80	41.38	**40.66**	46.35	46.96	**49.03**
Okapi-npn	**40.54**	**41.97**	40.49	46.54	46.61	48.97
Lnu-ltc	38.56	40.32	38.22	**48.73**	**47.31**	46.03
dtu-dtn	38.71	40.85	36.91	44.44	44.78	43.54
atn-ntc	37.21	38.47	40.50	42.91	43.99	48.56
ltn-ntc	34.47	36.12	36.65	42.47	43.11	42.94
ntc-ntc	25.74	27.45	26.52	32.73	33.46	35.64
ltc-ltc	26.93	29.26	25.91	37.27	38.34	37.72
lnc-ltc	27.46	29.67	29.28	36.93	39.18	37.21
bnn-bnn	20.21	22.33	25.79	17.95	15.17	20.06
nnn-nnn	11.87	12.06	12.70	13.85	13.21	14.83

language dependent), with no improvement for the English corpus (see Table 10), a small enhancement for the French collection (+0.5% from 51.64 to 51.91), yet an increase of 8.5% for the Spanish corpus (from a mean average precision of 48.85 to 53.02), and 9.4% for the Italian language (48.80 to 53.39). In Table 11, the improvement for the German collection is around 8.6% (words indexing, from 44.54 to 48.39) and of 15.5% for the Dutch corpus (from 46.86 to 54.14). As shown in Table 12, the enhancement is around 16% (from 39.80 to 46.17) for the Swedish collection, and of 13.7% with the Finnish language (from 46.35

Table 9. Mean average precision of various single searching strategies (Russian collection)

	Mean average precision			
Query TD Model	Russian words extended stemmer 28 queries	Russian words light stemmer 28 queries	Russian 5-gram 28 queries	Russian 4-gram 28 queries
Prosit	36.69	34.89	30.44	**34.43**
Okapi-npn	34.26	34.58	30.31	32.51
Lnu-ltc	36.34	**36.30**	27.36	29.75
dtu-dtn	32.67	32.95	28.49	30.55
atn-ntc	**37.06**	33.22	**31.29**	31.41
ltn-ntc	29.55	30.89	23.83	22.05
ntc-ntc	33.47	30.14	28.69	27.39
ltc-ltc	32.34	28.74	26.40	27.52
lnc-ltc	32.58	24.47	20.65	21.88
bnn-bnn	14.84	15.23	13.13	9.05
nnn-nnn	12.27	11.41	7.95	5.83

Table 10. Mean average precision using blind-query expansion

	Mean average precision			
Query TD Model	English 54 queries	French 52 queries	Spanish 57 queries	Italian 51 queries
doc=Okapi, query=npn	**48.83**	51.64	48.85	48.80
5 docs / 10 terms	48.79	51.33	52.74	52.97
5 docs / 15 terms	48.15	**51.91**	52.87	**53.39**
5 docs / 20 terms	47.37	51.30	**53.02**	52.35
10 docs / 10 terms	45.70	49.81	52.51	51.33
10 docs / 15 terms	44.10	48.59	52.55	51.17
10 docs / 20 terms	45.62	49.68	52.79	51.94

to 52.71). For the Russian corpus, the improvement shown in Table 13 is slight (+1.6% from 34.26 to 34.81).

6 Data Fusion

For the English, French, Spanish, Italian and Russian languages, we assumed that the n-gram indexing and word-based document representation approaches serve as distinct and independent sources of evidence regarding the content of documents. For the German, Dutch, Swedish and Finnish languages, we added the decompounding indexing approach in our documents (and queries) representation scheme.

Table 11. Mean average precision using blind-query expansion (German & Dutch collections)

	Mean average precision					
Query TD Model	German words 56 queries	German decomp. 56 queries	German 5-gram 56 queries	Dutch words 56 queries	Dutch decomp. 56 queries	Dutch 5-gram 56 queries
Okapi-npn	44.54	46.93	44.27	46.86	48.73	40.23
k doc. / m terms	5/10 46.46	5/10 50.32	5/50 **47.26**	5/10 52.32	5/10 54.60	5/100 43.12
	5/20 47.83	5/20 51.40	5/100 46.96	5/30 53.39	5/30 54.79	5/150 43.32
	5/40 **48.39**	5/50 **51.64**	5/125 46.88	5/50 **54.14**	5/40 **55.56**	5/200 **43.90**
	10/10 45.98	10/15 50.32	10/40 46.46	10/15 51.26	10/15 53.07	10/100 42.34
	0/15 46.31	10/30 50.20	10/100 46.50	10/20 51.14	10/20 52.81	10/150 42.67
	10/20 46.08	10/40 50.33	10/125 46.59	10/40 51.72	10/30 53.77	10/200 42.54

Table 12. Mean average precision using blind-query expansion (Swedish & Finnish collections)

	Mean average precision					
Query TD Model	Swedish words 54 queries	Swedish decomp. 54 queries	Swedish 4-gram 54 queries	Finnish words 45 queries	Finnish decomp. 45 queries	Finnish 5-gram 45 queries
Prosit	39.80	41.38	40.66	46.35	46.96	49.03
k doc. / m terms	3/20 **46.17**	3/10 **48.22**	3/30 42.48	3/20 52.50	3/10 52.03	3/15 50.98
	3/30 44.68	3/15 46.46	3/40 42.51	3/30 **52.71**	3/20 **53.37**	3/50 49.44
	3/60 42.76	3/40 43.73	3/50 **42.92**	3/40 50.04	3/30 52.93	3/125 49.06
	5/20 43.61	5/30 47.35	5/30 39.89	5/20 49.69	5/10 48.82	5/30 52.45
	5/30 44.12	5/40 46.80	5/40 41.53	5/30 47.90	5/15 47.85	5/60 **52.92**
	5/40 43.60	5/50 46.36	5/50 41.79	5/50 49.77	5/20 48.85	5/75 52.67

Table 13. Mean average precision using blind-query expansion (Russian collection)

	Mean average precision			
Query TD Model	Russian words extended stemmer 28 queries	Russian words light stemmer 28 queries	Russian 5-gram 28 queries	Russian 4-gram 28 queries
Okapi-npn	34.26	34.58	30.31	32.51
5 docs / 20 terms	**34.81**	32.68	29.27	30.76
5 docs / 30 terms	32.46	34.69	29.10	30.45
5 docs / 40 terms	31.87	**34.81**	29.64	30.62
10 docs / 20 terms	30.84	31.30	30.25	29.92
10 docs / 30 terms	29.24	33.00	30.07	30.17
10 docs / 40 terms	29.28	30.24	30.03	29.84
10 docs / 50 terms	27.99	28.88	29.32	29.46

Table 14. Data fusion combination operators

combMAX	$\max\ (\alpha_i \cdot RSV_k)$
combMIN	$\min\ (\alpha_i \cdot RSV_k)$
combSUM	$\sum\ (\alpha_i \cdot RSV_k)$
combANZ	$\sum\ (\alpha_i \cdot RSV_k)\ /\ \#ofnonzero(RSV_k)$
combNBZ	$\sum\ (\alpha_i \cdot RSV_k) \cdot (\#ofnonzero(RSV_k)$
combRSV%	$\sum\ (\alpha_i \cdot (RSV_k / MaxRSV^i))$
NormN	$\sum \left[\alpha_i \cdot \left[(RSV_k - MinRSV^i)/(MaxRSV^i - MinRSV^i)\right]\right]$

Table 15. Mean average precision using different combination operators ($\alpha_i = 1$, with blind-query expansion)

	Mean average precision				
Query TD	English	French	Spanish	Italian	Russian
Model #doc/#term	54 queries	52 queries	57 queries	51 queries	28 queries
Okapi-npn	0/0 48.83	10/10 49.81	10/10 52.51	10/20 51.94	10/20 31.30
Prosit	3/15 50.99	5/30 52.30	10/10 50.19	10/50 50.82	5/30 35.41
combMAX	48.83	52.27	50.19	50.82	35.41
combMIN	2.88	42.77	8.21	18.62	24.96
combSUM	51.13	53.58	51.89	51.87	**35.68**
combANZ	37.95	53.25	43.97	50.05	35.60
combNBZ	51.11	53.66	51.89	51.86	35.65
combRSV%	**53.60**	54.50	53.30	53.58	34.43
NormN	53.25	**54.69**	**53.49**	54.37	34.30
round-robin	50.24	52.61	53.16	**54.47**	34.11

Table 16. Mean average precision using different combination operators ($\alpha_i = 1$, with blind-query expansion)

	Mean average precision			
Query TD	German	Dutch	Swedish	Finnish
Model	56 queries	56 queries	54 queries	45 queries
Prosit word #doc/#term	5/20 48.40	10/20 51.14	3/60 42.76	5/30 47.90
Prosit decomp. #doc/#term	10/40 51.40	10/20 51.81	3/40 43.73	5/15 47.85
Prosit n-gram #doc/#term	5/175 49.46	10/150 44.23	3/40 42.51	3/125 49.06
combMAX	49.97	44.23	43.29	50.22
combMIN	35.54	6.30	33.80	33.36
combSUM	53.71	50.24	47.85	54.51
combANZ	47.85	31.90	41.32	49.25
combNBZ	53.70	50.81	47.57	**55.60**
combRSV%	54.46	53.99	48.23	54.49
NormN	**54.58**	**54.30**	**48.41**	54.16
round-robin	50.83	50.65	44.44	48.73

Table 17. Description and mean average precision (MAP) of our official runs

Run name	Query	Index	Model	Query expansion	Combined	MAP
FR	TD	word	Okapi	10 docs / 10 terms	round-	
UniNEfr	TD	word	Prosit	5 docs / 30 terms	robin	52.61
FR	TD	word	Okapi	10 docs / 10 terms		
UniNEfr2	TD	word	Prosit	5 docs / 30 terms	RSV%	**54.50**
SP	TD	word	Okapi	10 docs / 10 terms		
UniNEsp	TD	word	Prosit	10 docs / 10 terms	RSVnorm	**53.80**
SP	TD	word	Okapi	5 docs / 10 terms		
UniNEsp2	TD	word	Prosit	10 docs / 10 terms	RSVnorm	53.69
DE	TD	word	Prosit	5 docs / 20 terms		
UniNEde	TD	decomp.	Prosit	10 docs / 40 terms	RSVnorm	54.58
	TD	5-gram	Prosit	5 docs / 175 terms		
DE	TD	word	Pro+Oka	5 docs / 20 terms		
UniNEde2	TD	decomp.	Pro+Oka	10 docs / 40 terms	RSVsum	**56.03**
	TD	5-gram	Pro+Oka	5 docs / 175 terms		
IT	TD	word	Okapi	10 docs / 20 terms		
UniNEit	TD	word	Prosit	10 docs / 50 terms	RSV%	**52.23**
IT	TD	word	Okapi	10 docs / 20 terms		
UniNEit2	TD	word	Prosit	10 docs / 50 terms	RSVsum	51.56
NL	TD	word	Okapi	10 docs / 20 terms	round-	
UniNEnl	TD	decomp.	Okapi	10 docs / 20 terms	robin	**50.65**
	TD	5-gram	Prosit	10 docs / 150 terms		
NL	TD	word	Okapi	10 docs / 20 terms		
UniNEnl2	TD	decomp.	Okapi	10 docs / 20 terms	RSVsum	50.24
	TD	5-gram	Prosit	10 docs / 150 terms		
SV	TD	word	Pro+Oka	3 docs / 15 terms		
UniNEsv	TD	decomp.	Pro+Oka	3 docs / 15 terms	RSV%	48.53
	TD	4-gram	Pro+Oka	3 docs / 40 terms		
SV	TD	word	Pro+Oka	5 docs / 30 terms		
UniNEsv2	TD	decomp.	Pro+Oka	5 docs / 50 terms	RSVnorm	**49.03**
	TD	4-gram	Pro+Oka	5 docs / 30 terms		
FI	TD	word	Prosit	5 docs / 30 terms		
UniNEfi	TD	decomp.	Prosit	5 docs / 15 terms	RSVsum	**54.51**
	TD	5-gram	Prosit	3 docs / 125 terms		
FI	TD	word	Prosit	5 docs / 30 terms		
UniNEfi2	TD	decomp.	Prosit	5 docs / 15 terms	RSVsum	53.55
	TD	5-gram	Prosit	3 docs / 125 terms		
RU	TDN	word	Okapi	10 docs / 20 terms		
UniNEru	TDN	word	Prosit	5 docs / 30 terms	RSVsum	35.32
RU	TD	word	Okapi	10 docs / 20 terms		
UniNEru1	TD	word	Prosit	5 docs / 30 terms	RSVsum	31.83
RU	TD	5-gram	Okapi	10 docs / 50 terms		
UniNEru2	TD	5-gram	Prosit	5 docs / 40 terms	RSVsum	**32.77**
	TD	4-gram	Okapi	10 docs / 50 terms		
	TD	4-gram	Prosit	5 docs / 40 terms		
RU	TDN	word	Okapi	10 docs / 10 terms		
UniNEru3	TDN	word	Prosit	5 docs / 20 terms	RSVsum	**42.24**

In order to combine these two and three point indexing schemes respectively, we evaluated various fusion operators, as suggested by Fox and Shaw [6]. Table 14 shows their precise description. For example, the combSUM operator indicates that the combined document score (or the final retrieval status value) is simply the sum of the retrieval status value (RSV_k) of the corresponding document D_k computed by each single indexing scheme. CombNBZ specifies that we multiply the sum of the document scores by the number of those retrieval schemes able to retrieve the corresponding document. In Table 14, we can see that both the combRSV% and NormN apply a normalization procedure when combining document scores. When combining the retrieval status value (RSV_k) for various indexing schemes, we may multiply the document score by a constant α_i (usually equal to 1) in order to attribute a different weight to each retrieval scheme according to its overall performance. In addition to using these data fusion operators, we also considered the round-robin approach, where in turn we take one document from all individual lists and remove duplicates, keeping the most highly ranked instance.

Table 15 and Table 16 show the evaluation of various data fusion operators, comparing them to the single approach using the Okapi and the Prosit probabilistic models. As shown in these tables, the NormN or combRSV% fusion strategies usually improve retrieval effectiveness over the best single retrieval model.

7 Conclusion

In this fourth CLEF evaluation campaign, we proposed a general stopword list and stemming procedure for eight European languages. Currently it is not clear if a stemming procedure such as the one we suggested, where only inflectional suffixes are removed from nouns and adjectives, could produce better retrieval effectiveness than a stemming approach that takes both inflectional and derivational suffixes into account. We also suggested a simple decompounding approach for German, Dutch, Swedish and Finnish. In order to achieve better retrieval performance, we used a data fusion approach, one requiring that document (and query) representation be based on two or three indexing schemes.

Acknowledgments. The author would like to thank C. Buckley from SabIR for giving us the opportunity to use the SMART system. This research was supported by the Swiss National Science Foundation (grant #21-66 742.01).

References

1. Amati, G., Carpineto, C., Romano, G.: Italian Monolingual Information Retrieval with PROSIT. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.): Advances in Cross-Language Information Retrieval. Lecture Notes in Computer Science, Vol. 2785. Springer-Verlag, Berlin Heidelberg New York (2003) 257–264
2. Amati, G., van Rijsbergen, C.J.: Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. ACM Transactions on Information Systems, 20 (2002) 357–389.

3. Braschler, M., Ripplinger, B.: Stemming and Decompounding for German Text Retrieval. In Proceedings 25th European Conference in IR. Lecture Notes in Computer Science, Vol. 2633. Springer-Verlag, Berlin Heidelberg New York (2003) 177–192
4. Buckley, C., Singhal, A., Mitra, M., Salton, G.: New Retrieval Approaches Using SMART. In Proceedings TREC-4. NIST Publication #500-236, Gaithersburg (1996) 25–48
5. Chen, A.: Cross-Language Retrieval Experiments at CLEF 2002. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.): Advances in Cross-Language Information Retrieval. Lecture Notes in Computer Science, Vol. 2785. Springer-Verlag, Berlin Heidelberg New York (2003) 28–48
6. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In Proceedings TREC-2. NIST Publication #500-215, Gaithersburg (1994) 243–249
7. Kraaij, W., Pohlmann, R.: Viewing Stemming as Recall Enhancement. In Proceedings of the ACM-SIGIR'96. The ACM Press, New York (1996) 40–48
8. Lovins, J.B.: Development of a Stemming Algorithm. Mechanical Translation and Computational Linguistics 11 (1968) 22–31
9. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14 (1980) 130–137
10. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a Way of Life: Okapi at TREC. Information Processing & Management 36 (2000) 95–108
11. Savoy, J.: Report on CLEF 2002 Experiments: Combining Multiple Sources of Evidence. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.): Advances in Cross-Language Information Retrieval. Lecture Notes in Computer Science, Vol. 2785. Springer-Verlag, Berlin Heidelberg New York (2003) 66–90
12. Singhal, A., Choi, J., Hindle, D., Lewis, D.D., Pereira, F.: AT&T at TREC-7. In Proceedings TREC-7. NIST, Publication #500-242, Gaithersburg (1999) 239–251