The LIC2M's CLEF 2003 System

Romaric Besançon, Gaël de Chalendar, Olivier Ferret,
Christian Fluhr, Olivier Mesnard and Hubert Naets
CEA/LIST - LIC2M
{*Romaric.Besancon,Gael.de-Chalendar,Olivier.Ferret,*
Christian.Fluhr,Olivier.Mesnard, Hubert.Naets}*@cea.fr*

Abstract

For its first birthday, the LIC2M has participated to the Small Multilingual Track of CLEF 2003. Our system is based on a deep linguistic analysis of documents and queries and on an original search algorithm inherited from the Spirit (EMIR) system. With a partially developed system, we obtained average results that will serve as a baseline for us in our future participations to IR evaluations.

1 Introduction

CLEF 2003 is the first participation of the LIC2M (Multimedia and Multilingual Knowledge Engineering Laboratory) to an international evaluation campaign. The lab was created inside the CEA (Commissariat à l'Énergie Atomique) at the beginning of 2002 and has inherited from the technology developed during the last three decades in the CEA with the EMIR [4] and SPIRIT [5] systems.

As we decided to restart the development from scratch to use at best this technological basis but with modern software engineering methods and tools, the system that was used for the Small Multilingual Track of CLEF 2003 has been developed in a short period of time and is still incomplete: some important modules are not included (see section 3) and the overall system needs tuning and validation. However, we obtained this year results similar to those of the previous version of the system that participated to TREC7 (cf. section 7). The future developments and tuning of the system (section 8) should then lead to better results in future evaluations.

The paper is organized as follow: we begin by an overview of the system (section 2) followed by sections 3 to 6 dedicated to the description of its components (Linguistic processing, Indexing, Query Processing and Search and Merging). Next, we describe our results (section 7) before presenting our conclusions and the plans for our future work (section 8).

2 Outline of the process

The general architecture of the process is presented in figure 1. The system is composed of four elements :

- a linguistic analyzer, which is used to process documents and queries;
- an indexer to store the documents in indexes, as a matrix documents×terms;
- a query processor that reformulates the query to suit the search (monolingual and multilingual reformulations);
- a search engine that searches the indexes for the closest documents to the reformulated queries and merges the results obtained for each language.

The linguistic analyzer is itself composed of a set of modules customized with resources. The combination of modules and resources (dictionaries, set of categories..) depends on the language.

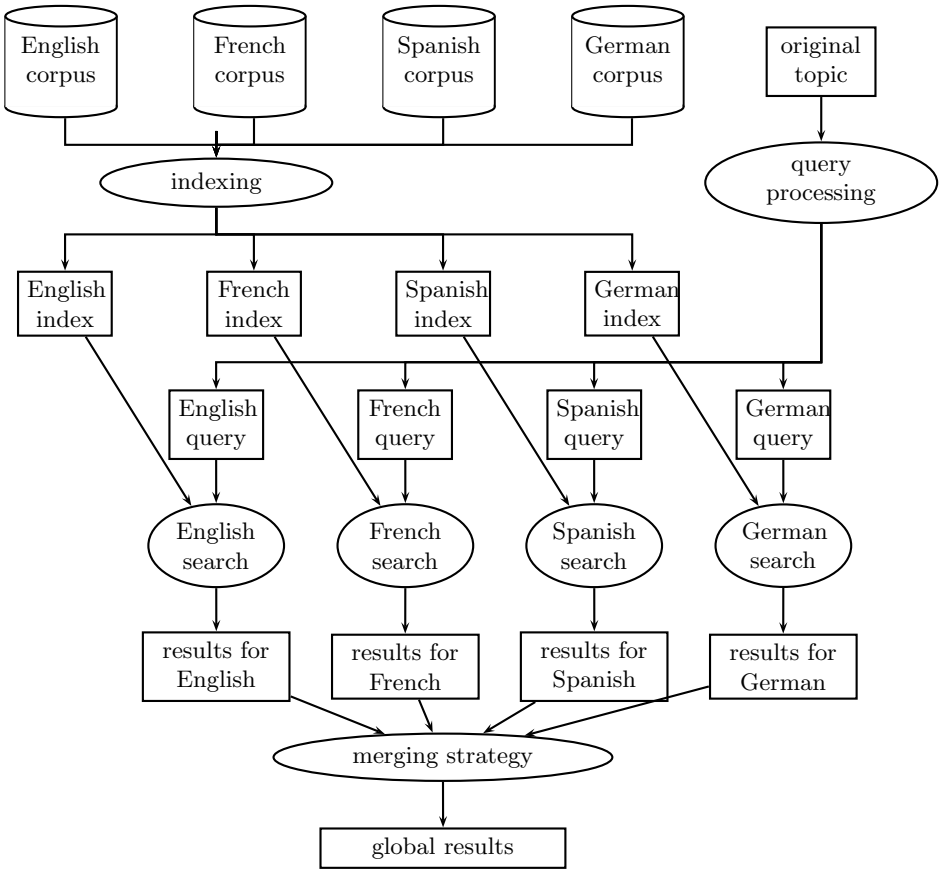

Figure 1: General architecture of the search

The indexing process can be described as follows: each document from a collection is first processed through the linguistic analyzer, according to the language it is written in. The output of the linguistic analyzer is a stream of tokens, each token being a set of three pieces of information: lemma[1], category and position. Position means offset of the word relative to the beginning of the document. The pair (lemma, category) is considered as a *term*. Each distinct pair is used as an entry in the index. The indexer stores for each entry, the documents containing the term, and the positions of the terms in each document. Four indexes are built, one for each language.

The query process performs the following tasks: the topic is processed through the linguistic analyzer; pairs (lemma, category) from the stream of tokens are considered at this stage as *concepts* and concepts are reformulated into four sets of terms, one for each target language, using monolingual and multilingual reformulation dictionaries.

The search engine then sends requests to the four indexes to get a list of the documents containing the largest number of query terms, and merges the four lists obtained. The search engine takes into account the original concepts and their weights to score the document.

3 Linguistic Processing

The linguistic processing is a fundamental part of the system. We decided to base all the processing of our system on a deep linguistic analysis designed to extract precise information from the structure of texts. We will see below that this goal is not yet fully reached since all modules are not completely implemented or fully functional.

[1]Here and in the rest of the paper, a lemma is a normalized form and can group several lemmas (in the linguistic sense) of strict synonyms.

Figure 2 shows the organization of the modules in the linguistic processing chain. The input is a raw text in a given language[2] converted to Unicode.

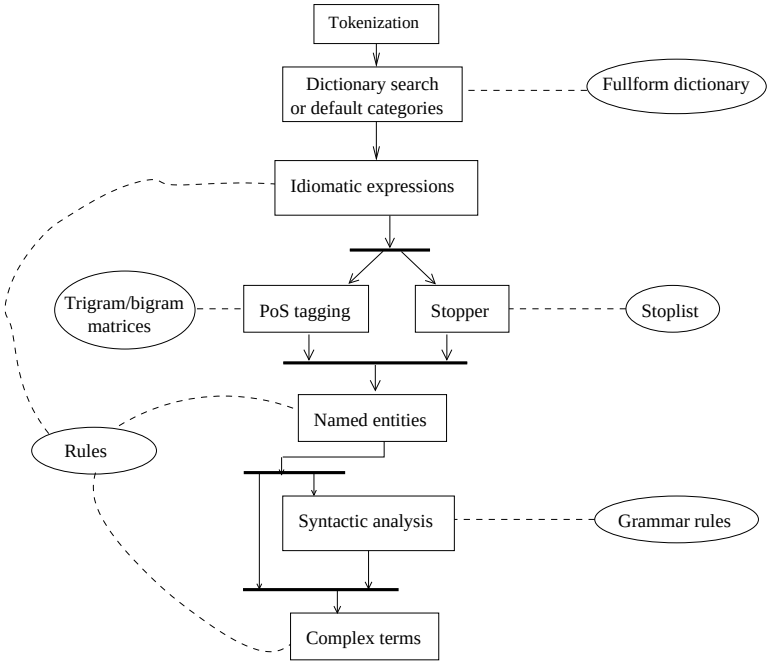

Figure 2: Overview of the linguistic processing

The first step is the tokenization of the text in order to find word and sentence breaks. Next, tokens are searched in a full form dictionary. If they are found in the dictionary, they are associated to their lemmas and are given all the morphosyntactic categories associated with this form.

It should be noted that our system uses a specific set of morphosyntactic categories that makes it quite different from other systems: the categories are positional, meaning that the category itself allows to distinguish which words can appear before or after another words. For example, for French, there are pre-nominal adjectives and post-nominal adjectives. The positional properties of this large set of categories (137 for French, 120 for English, 100 for Spanish and currently only 38 for German) will allow a very effective disambiguation, as we will see below.

If a token is not found in the dictionary, it is given a default set of morphosyntactic categories based on some typographical properties: a token beginning with an uppercase letter will obtain the categories of proper nouns, for example.

After tokenization, the idiomatic expressions are detected and replaced by a single token. Idiomatic expressions are phrases or usual compound nouns that are in our dictionary. They can be non-contiguous like phrasal verbs: "*switch the light on*" is replaced by "*switch-on the light (on)*" (the information about the preposition is kept because it will be used during syntactic analysis). The detection of idiomatic expressions is performed by applying a set of rules, that are triggered on specific words and tested on left and right contexts of the trigger (note that the same technology is used for identifying named entities and compounds).

After these steps, each token have several possible morphosyntactic categories. The goal of the part-of-speech tagger is to drastically reduce the number of possible categories for each word. We will see below that our technology should be able to obtain results at the level of state of the art systems without any statistics. Unfortunately, this module was not ready at the time of the

[2]Our language and encoding identifier has not been used for CLEF as the language and encoding of each given corpus are already known. Even if some texts of one language's corpus are actually in another language (for example, some LATimes texts are in Spanish), we have decided to ignore this point.

CLEF campaign and thus we only remove functional and too frequent words with stoplists. We then kept all other tokens with content word categories.

The fifth step uses the same algorithms as the idiomatic expressions step, with a specific set of rules, to extract the named entities like people or locations names. The extracted tokens groups are replaced by a single token. This module has quite good results with a precision 80% and a recall around 60% depending on the language and the entity type. This evaluation has been done for English (5,000 texts), French (5,000 texts) and Spanish (50 texts) using a set of manually annotated texts.

The resulting text should then be used to do a syntactic analysis Again, this module was not ready at the time of CLEF and thus was not used. We will see in section 8 that we hope to improve our results in a future evaluation by using all the modules our system is designed to work with.

The last step of the linguistic analysis is the extraction of compound nouns. These compounds are important indicators for information extraction but also for a lot of other tasks. This module also relies on the rule-bases technology used to extract idiomatic expressions and named entities. Since no syntactic analysis is performed, every compound corresponding to a pattern (like *Noun Preposition Noun*) described in a rule is kept without any tense or genre agreement checking.

The results of the linguistic processing that we described above contain linguistic data of various levels that are used as well for the documents indexing (section 4) and for the queries processing (section 5).

4 Indexing

The search engine described in section 6 relies on basic functions which give access to raw information on the collection. Efficient implementation of these basic functions allow an easy use of the global system. We built inverted files (containing, for each term found in the collection of documents, a list of all documents containing this term), using the Lemur toolkit (V2.01) [7, 1]. Within this context, our design choices consist in:

1. considering terms as a pair of two complementary pieces of information: the lemma of the word and its grammatical category as found in the document. This was possible thanks to the deep linguistic processing described in the previous section. The first benefit is that access to entries in indexes and compression of indexes are more efficient because there is less entries than if we had considered inflected forms of terms. The second benefit is that we take into account the semantic of terms at this very basic layer of the system which lead to more simple reformulation calculation in the query processing layer. Adding the grammatical category must lead to more discriminant inverted files than relying only on lemma, but we did not have the opportunity to evaluate the impact of such distinction. Only macro category have been considered: Noun, Verb, Adjective and Proper Noun. Entries of indexes are built doing concatenation of the lemma and a number which represents the category (the cost is then only of two characters for each entry).

2. indexing every terms, without any frequency filtering, using only "*stop words*" lists. Indexing a large number of terms does not cost a lot when efficient compression schemes are used. In the future version, the indexing terms will be chosen based on their part-of-speech (keeping only content words such as nouns, verbs adjectives). Stoplists will then be used only as a complement to eliminate some non-relevant content words. In the current version the stoplists represent approximately 60% of the text.

3. building separate indexes for each language: English, French, Spanish and German. There is no drawback because the language is identified for each text and we consider that documents are monolingual (only one language per document), and this leads to more manageable indexes.

The statistics of the index results are given for each corpus in table 1, indicating for each language the size of the corresponding corpus, the number of documents in the corpus, the total number of terms that where indexed (including single terms and compounds), the number of distinct single terms indexed, the number of distinct compounds indexed, the total size of the index files (as kept on disk), and the memory size that is used to load the index (includes term and document lists and lookups on the index files). Because indexes are too big to fit in memory, all information is not loaded but stays in files. Looks up are used to hold the offset of each entry.

	corpus size (Mo)	nb docs	nb terms	nb distinct uniterms	nb distinct compounds	index size (Mo)	memory size (Mo)
fre	326	129.806	30.756.020	297.084	2.281.349	476	185
eng	576	169.477	48.387.519	512.196	1.571.058	593	136
ger	632	287.670	55.605.358	1.076.714	179.357 [3]	603	89
spa	1.084	454.045	112.603.093	747.755	3.012.136	1.325	261

Table 1: Statistics of the indexing process

It can be can noted that the total size of indexes is larger than the original text (except for German). Our indexes are not very compact because of the great number of compound terms (which increases the size of vocabulary and add many entries in indexes). We have noted that we actually need 800 Mo to load the four indexes and run a request.

In future versions of our system, we will replace the Lemur toolkit with a more efficient implementation, allowing a smaller memory footprint to store the vocabulary, a transaction model to manage both read and write access (to index new files while executing retrieval request) and handling of Unicode text.

5 Query Processing

5.1 Linguistic processing

Each query is first processed through the linguistic analyzer corresponding to the query language, as described in section 3. The three fields of the query (title, description, narrative) are kept for this analysis.

The result is a query composed of a list of elements that can be a lemma, associated with its part-of-speech (limited to nouns, verbs, and adjectives); a named entity, associated with its type; a compound, in a normalized form, associated with its part-of-speech[4].

The query is then filtered using a stoplist containing meta-words (words used in the narrative to describe what are relevant documents, such as : "document", "relevant" etc.). These meta-words stoplists have been built on the basis of CLEF 2002 topics, from a first selection using frequency information, and revised manually.

No deep semantic analysis has been performed on the narrative, to take into account, for instance, the negative descriptions of the topics ("*documents that contain ... are not relevant*"). Some negative descriptive elements have then been kept in the final queries.

After this processing, the query is a list of indexing elements, in the original language of the topic. These elements are called the *concepts* of the query.

5.2 Query reformulation

The list of query concepts is augmented with additional knowledge, using external resources (such as monolingual and bilingual reformulation dictionaries) and using the corpus to search as a reference (the indexes are used to filter out words inferred by the reformulation).

[4]The system still being under development, the compounds were not properly taken into account, especially in query expansion and translation, and were just used for monolingual search.

The reformulated query is then a list of the original query terms, called *query concepts*, and a list of inferred terms, called *search terms*. Each search term is linked to a query concept, and a weight is associated to the link. An example of query reformulation is presented in figure 3 at the end of this section.

5.2.1 Translation

In order to query a corpus in a language different from the original query language, a translation of the query terms is performed, using bilingual dictionaries. Each term of the query is translated into several terms in target language. The translated words form the search terms of the reformulated query. The links between the search terms and the query concepts can also be weighted by a confidence value (between 0 and 1) indicating the confidence in the translation. In this first version, all translations were assigned the same weight.

In the small multilingual task, the only language pair for which we did not have a bilingual dictionary is the Spanish/German pair. For this pair, we used a two-step translation, using a pivot language (terms are first translated from topic language to the pivot language, and then from pivot language to target language). In this case, the confidence in the translation is the product of the confidences of the two successive translations. For the submitted runs, we used only one pivot language for the Spanish/German pair (chosen language was French), but a concurrent use of different pivot languages could also be used for this translation.

5.2.2 Monolingual reformulation

A semantic expansion was also performed to increase the lexical variety of the query concepts, using monolingual reformulation dictionaries (containing mostly synonymy information).

In the runs we submitted, this semantic expansion was only performed for monolingual query expansion. In a more general approach for crosslingual retrieval, several combinations of multi-step translation and monolingual expansion can be imagined. We plan to test several of them on the CLEF 2003 data.

5.2.3 Topical expansion

The semantic expansion described in the previous section is mainly based on synonyms. [8] shows that this kind of expansion reliably improves results if the terms of the queries are semantically disambiguated. As we did not perform such a disambiguation, we chose to reinforce the representation of the context of each query by adding to it words that are topically linked to its words after the semantic expansion step[5].

The selection of such words is based on a network of lexical cooccurrences. For French, the only language for which we tested this kind of expansion, this network was built from a 39 million word corpus made of 24 months from the *Le Monde* newspaper (see [3] for more details). After a filtering procedure was applied [2] to select the cooccurrences that are likely to be supported by a topical relation, we got a network of 7,200 lemmas and 183,000 cooccurrences.

This network is used in a three-stage process that relies on a kind of bootstrapping. First, a set of words from the network that are strongly linked to the considered query are selected. The strength of this link is set by the number of words of the query the word from the network is linked to (3 words in our experiments). Most of these words, which are called expansion words, are topically close to the query but some of them also represent noise. The next stage aims at discarding this noise. It consists in selecting the words of the query that are the most representative of its topic. This selection is based on the words resulting from the first stage: we assume that a query word is a significant one if it has contributed to the selection of a minimal number of expansion words (2 words in our experiments). The final stage is identical to the first one, except that the expansion is done from the selected words of the query and not from all its plain words.

[5] According to [6], topical relations are non systematic semantic relations such as the ones between tennis, racket, ball and net for instance.

Moreover, the number of expansion words is arbitrarily set to 10 to avoid swamping the initial words of the query.

This topical expansion was applied to the 60 French topics of CLEF 2003. A set of expansion words was produced for 42 of them. This set was empty for the other ones, which means that it was not possible in these cases to build from the network of cooccurrences a significant representation of the topic of the query. As an example, the result of the topical expansion of the topic C164, *Les condamnations pour trafic de drogue en Europe (European Drug Sentences)*, is the following list of words: {*amende, infraction, prison, délit, procès, pénal, crime, juge, cocaïne, sursis*}.

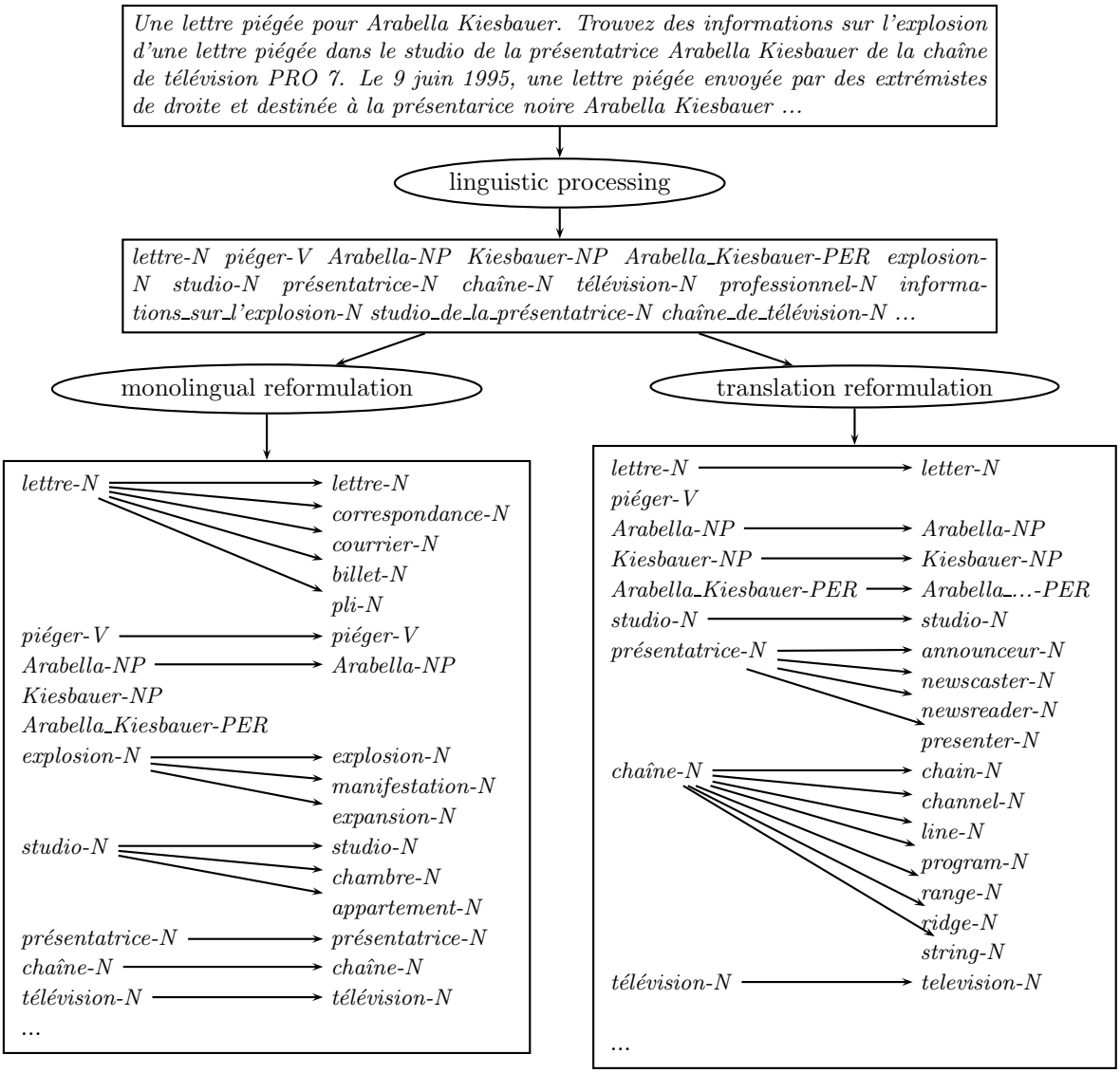

Figure 3: Summarization of the query construction on an example

6 Search and Merging

The original topic is transformed, during the query processing, into four different queries, one for each language. The search is performed independently for each of theses queries on the index of the corresponding language. 1,000 documents are retrieved for each language. The 4,000 retrieved

documents from the four corpora are then merged and sorted by their relevance to the topic. Only the first 1,000 are then kept. We present in the following sections the search technique and the merging strategy.

6.1 Search

For each expanded query, the search is performed by retrieving from the index of the corresponding language, for each search term of the query, the documents containing the term. A term profile is then associated to each document: this profile consists in a binary vector of size the number of search terms of the query, each component of which indicating the presence or absence of the search terms in the document. The frequency of the term in the document is not used: we consider that if a document contains relevant information with respect to a topic, then the document is relevant, event if the document contains other material that is not relevant to the topic (a visualization step can then select the relevant parts and show them to the user). However, further version of the search engine will also be able to take the frequency of the term in the document into account, to make the search engine more suitable to standard evaluation procedures.

Since we kept, in the expanded query, the links between the search terms and the query concepts, we can associate to each document a concept profile, indicating the presence/absence of each query concept in the documents.

The retrieved documents are then classified, grouping in the same cluster the documents that share the same concept profile. This classification is motivated by at least two reasons: it makes it easy to merge the results from different languages (see following section) and the visualization of the results is more efficient: the clustering of the results, and the association of a concept profile to a cluster makes it easier for the user to search the results (a cluster corresponding to a non-relevant subset of query concepts can simply be ignored).

6.2 Merge and sort

Given that a concept profile is associated to each class, the merging strategy is quite straightforward: since the concepts are in the original query language, the concept profiles associated to the classes are comparable, and the classes having the same profile are simply merged.

The classes are then sorted by their relevance to the query. For this purpose, we use the *idf* (inverse document frequency) weights of the terms, defined for a term t by the formula $idf(t) = \log \frac{N}{df(t)}$, where $df(t)$ is the document frequency of the term (*i.e.* the number of documents containing the term) and N is the total number of documents in the corpus.

The first idea is to use the *idf* weights of the terms in each language, and compute the weight of a concept by some combination of the weights of the terms derived from the concept, and then associate a weight to a concept profile derived by the concepts it contains. However, in this case, the weights computed for the different languages are not comparable (*idf* of the terms depend of the corpora).

We decided to compute a crosslingual pseudo-*idf* weight of the concepts, using only the corpus composed of the 4000 documents kept as the result of the search. The *idf* weight of the concepts is computed on this corpus, using only information contained in the concept profiles of the classes and the size of the classes. A weight can then be associated to each concept profile by computing the sum of the weights of the concepts present in the profile.

The classes are then sorted by their weights: all documents in a class are given the weight of the class (the documents are not sorted inside the classes). The list of the first 1000 documents from the best classes is then built and used for the evaluation.

We used this simple weighting procedure, but a lot of other procedures can be imagined. We plan to test several more sophisticated weighting schemes, using in particular the document frequency of the search terms and the confidence weight in the expansion associations.

7 Results

Figure 4 shows the precision-recall curves of our five runs. There are few differences between them except for the Spanish run that is slightly better than the others. One probable reason for that is that the number of relevant documents for Spanish (2,368) is largely greater than in the other languages (1,825 in German, 1,006 in English and 946 in French).

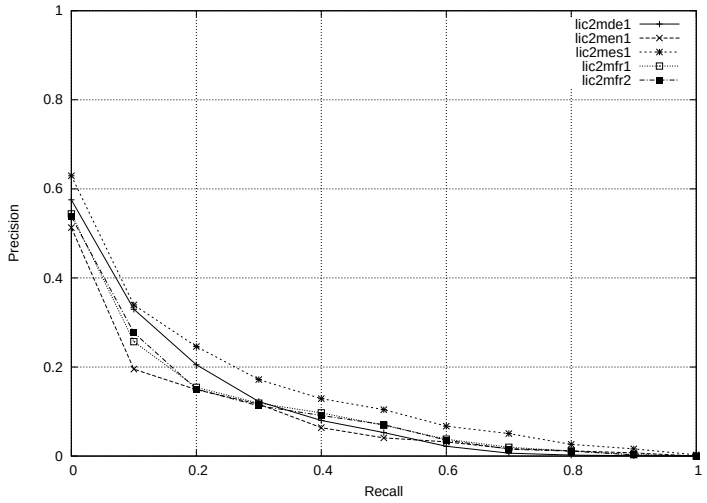

Figure 4: Results of the 5 runs

Table 2 details the numbers and percentages of relevant documents found in each language for each run. The percentages are the percentages of relevant documents found among all the relevant documents for a given language (cf. previous paragraph for these numbers).

	number (%) of relevant found			
interro. lang.	fre	eng	spa	ger
fre (run 1)	409 (43)	310 (31)	204 (9)	620 (34)
fre (run 2)	631 (67)	154 (15)	201 (8.5)	565 (31)
eng	354 (37)	546 (54)	252 (23)	274 (15)
spa	257 (27)	303 (30)	948 (40)	265 (15)
ger	333 (37)	139 (14)	115 (5)	1,060 (58)

Table 2: Percentages of relevant documents of each language found depending on the interrogation language

As expected, our system exhibits its best results for monolingual retrieval. It shows that we still have work to do on our reformulation process and its resources. Concerning the resources, we see that the French-Spanish dictionary, used to reformulate between French and Spanish but also as a pivot language for Spanish to German and German to Spanish, needs particular attention as the number of relevant documents retrieved when using it is very low.

Concerning monolingual reformulation, we submitted two runs for French topics: one with only semantic expansion of queries and the other one with semantic and topical expansion of queries. The results for these two runs were globally very similar: in the first case, 1543 relevant documents were returned with a R-precision of 0.1425 while in the second case, 1551 relevant documents were returned with a R-precision of 0.1438. However, the two sets of relevant documents are not identical since their intersection only contains 1305 documents. Topical expansion brings new relevant documents but also discards relevant documents brought by semantic expansion in the

same proportion. More precisely, Table 2 shows that it favors monolingual retrieval while it has a negative impact on crosslingual retrieval.

Considering this observation, we tested if a combination of the two runs could lead to improve results. We adopted a basic strategy: the two lists of documents were interleaved in the decreasing order of their score; only one occurrence of each document was kept and the resulting list was cut at 1000 documents. 1730 relevant documents were returned with a R-precision of 0.1443. From the viewpoint of the number of relevant documents, the benefit is low (+179 documents, *i.e.* an increase of 11.6%) but not insignificant. Moreover, most of the relevant documents that could be caught by our combination of runs were actually caught. However, from the viewpoint of R-precision, the benefit is not significant, which probably means that the rank of most of the new relevant documents is high.

8 Conclusion and Future Work

Despite the numerically modest results of our system, our participation to CLEF'2003 is an encouragement for the future. Indeed, in a short period we implemented a tokenizer, a morphosyntactic analyzer, a named entities recognizer, an indexer and a search engine. All this work was done with a multilingual methodology that necessitated to gather several monolingual and bilingual resources. However, our system still requires more work both on the technology and the resources.

Some work has already been done: we have implemented a new method to store our dictionaries. A first version of the morphosyntactic tagger works but will need to be replaced for a more speed-efficient one. Some other work have started: we are currently working on the syntactic analyzer that will extract nominal and verbal chunks and find dependencies inside the chunks and between them. This last part will allow to find compound nouns with better precision that our current method. These compound nouns are a fundamental part of our search algorithm and thus their current lack probably takes a great part in our current poor results. They will also permit to have a better multilingual search due to the noise introduced by a word by word translation. In the next year, we will also extend our languages coverage to three new languages and introduce monolingual and crosslingual relevance feedback.

References

[1] The Lemur Toolkit for language modeling and information retrieval. http://www-2.cs.cmu.edu/ lemur/.

[2] Olivier Ferret. Filtrage thématique d'un réseau de collocations. In *TALN 2003*, pages 347–352, 2003.

[3] Olivier Ferret and Brigitte Grau. A bootstrapping approach for robust topic analysis. *Natural Language Engineering*, 8(2/3):209–233, 2002.

[4] Christian Fluhr, Patrick Mordini, André Moulin, and Erwin Stegentritt. Emir final report. Technical Report ESPRIT project 5312, DG III, Commission of the European Union, CEA, oct 1994.

[5] Christian FLuhr, Dominique Schmit, Philippe Ortet, Faiza Elkateb, and Karine Gurtner. Spirit-w3, a distributed crosslingual indexing and retrieval engine. In *INET'97*, june 1997.

[6] M.A.K. Halliday and R. Hasan. *Cohesion in English*. Longman, London, 1976.

[7] Paul Ogilvie and James Callan. Experiments using the lemur toolkit. In *Proceedings of TREC-2001, The Tenth Text REtrieval Conference*, pages 103–108, 2001.

[8] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, IE, 1994.