

Exploiting Lexical Expansions and Boolean Compositions for Web Querying

Bernardo Magnini and Roberto Prevete

ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive
38050 Povo (TN), Italy,
{magnini | prevete}@irst.itc.it
<http://ecate.itc.it:1024/projects/question-answering.html>

Abstract

This paper describes an experiment aiming at evaluating the role of NLP based optimizations (i.e. morphological derivation and synonymy expansion) in web search strategies. Keywords and their expansions are composed in two different Boolean expressions (i.e. expansion insertion and Cartesian combination) and then compared with a keyword conjunctive composition, considered as the baseline. Results confirm the hypothesis that linguistic optimizations significantly improve the search engine performances.

Introduction

The purpose of this work was to verify if, and in which measure, some linguistic optimizations on the input query can improve the performance of an existing search engine on the web¹.

First of all we tried to determine a proper baseline to compare the optimized search strategies. Such a baseline should reflect as much as possible the average use of the search engine by typical users when querying the web. A query is usually composed of a limited number of keywords (i.e. two or three), in a lemmatized form, that the search engine composes by default in a conjunctive

expression. Starting from this level (we call it “basic level”) we have designed two more sophisticated search strategies that introduce a number of linguistic optimizations over the keywords and adopt two composition modalities allowed by the “advanced search” capabilities of the search engine. One modality (i.e. Keyword expansion Insertion Search - KIS) first expands each keyword of the base level with morphological derivations and synonyms, then it builds a Boolean expression where each expansion is added to the base keyword list. The second modality (i.e. Keyword Cartesian expansion Search - KCS) adopts the same expansions of the previous one, but composes a Boolean expression where all the possible tuples among the base keywords and expansions are considered.

The working hypothesis is that the introduction of lexical expansions should bring an improvement in the retrieval of relevant documents. To verify the hypothesis, a comparative evaluation has been carried out using the three search modalities described above over a set of factual questions. The results of the queries have been manually scored along a five value scale, with the aim of taking into account not only the presence in the document of the answer to the question, but also the degree of contextual information provided by the document itself with respect to the question. Both the presence of the answer and the contextual information have been estimated by two relevance functions, one that considers the document position, the other that does not.

The experiment results confirm that the introduction of a limited number of lexical expansions (i.e. 2-3) improves the engine performance. In addition, the Cartesian

¹ The results reported in this paper are part of a more extended project under development at ITC-irst, which involves a collaboration with Kataweb, an Italian web portal. We thank both Kataweb and Inktomi Corporation for kindly having placed the search engine for the experiments at our disposal.

composition of the expansions behaves significantly better than the search modality based on keyword insertion.

Some of the problems that we faced with in this work have been already discussed in previous works in the literature. The use of query expansions for text retrieval is a debated topic. Voorhees (1998) argues that WordNet derived query expansions are effective for very short queries, while they do not bring any improvements for long queries. From a number of experiments (Mandala et al., 1998) conclude that WordNet query expansions can increase recall but degrade precision performances. Three reasons are suggested to explain this behavior: (i) the lack of relations among terms of different parts of speech in WordNet; (ii) many semantic relations are not present in WordNet; (iii) proper names are not included in WordNet. (Gonzalo et al., 1998) pointed out some more weaknesses of WordNet for Information Retrieval purposes, in particular the lack of domain information and the fact that sense distinctions are excessively fine-grained for the task. A related topic of query expansion is query translation, which is performed in Cross-Language Information Retrieval (Verdejo et al. 2000).

This work brings additional elements in favor of the thesis that using linguistic expansions can improve IR in a web search scenario. In addition we argue that, to be effective, query expansion has to be combined with proper search modalities. The evaluation experiment we carried out, even within the limitations due to time and budget constraints, was designed to take into account the indications that came out at the recent TREC workshop on Question Answering (Voorhees, 2000).

The paper is structured as follows. Section 1 and 2 respectively present the modalities for the linguistic expansion and for the query composition. Section 3 reports the experimental setting for the comparative evaluation of the three search modalities. Section 4 describes and discusses the results obtained, while in the conclusions we propose some directions for future work.

1 Lexical expansion

Two kinds of lexical expansion have been used in the experiment: morphological derivations and synonym expansions. Both of them try to expand a “basic-keyword”, that is a keyword directly derived from a natural language question. The language used in the experiments is Italian.

1.1 Basic keywords

The idea is that this level of keywords should reflect as much as possible the words used by an average user to query a web search engine. Given a question expressed with a natural language sentence, its basic keywords are derived selecting the lemmas for each content word of the question. Verbs are transformed in their corresponding nominalization. Furthermore we decided to consider collocations and multiwords as single keywords, as most of the currently available search engines allow the user to specify “phrases” in a very simple way. In the experiments presented in the paper multiword expressions are manually recognized and then added to the basic keyword list.

Figure 1 shows a couple of questions with their respective basic keywords.

NL-QUESTION: Chi ha inventato la luce elettrica? (*Who invented the electric light?*)
BASIC-KEYWORDS: inventore (*inventor*)
luce_elettrica (*electric_light*)

NL-QUESTION: Quale è il fiume più lungo del mondo? (*Which is the longest world river?*)
BASIC-KEYWORDS: fiume (*river*) più_lungo (*longest*) mondo (*world*)

Figure 1: Basic keywords extraction from questions.

1.2 Morphological derivation

Morphological derivations are considered because they introduce new lemmas that we might find in possible correct answers to the question, improving in this way the engine recall. For instance, for a question like “Chi ha inventato la luce elettrica?” (“*Who invented the electric light?*”) we can imagine different contexts for the correct answer, such as “la luce elettrica fu inventata da Edison” (“*Electric light*

was invented by Edison”), “L’inventore della luce elettrica fu Edison” (“*The inventor of electric light was Edison*”), “L’invenzione della luce elettrica è dovuta a Edison” (“*The invention of electric light is due to Edison*”), where different morphological derivations of the same basic keyword “inventore” (“*inventor*”) appear. Derivations have been automatically extracted from an Italian monolingual dictionary (Disc, 1997), and collected without considering the derivation order (i.e. “inventare” belongs to the derivation set of “inventore” even if in the actual derivation it is the noun that derives from the verb).

1.3 Synonyms

Keyword expansion based on synonyms can potentially improve the system recall, as the answer to the question might contain synonyms of the basic keyword. For instance, the answer to the question “Chi ha inventato la luce elettrica?” (“*Who invented the electric light?*”) might be one among “Lo scopritore della luce elettrica fu Edison” (“*The discoverer of electric light was Edison*”), “L’inventore della illuminazione elettrica fu Edison” (“*The inventor of electric illumination was Edison*”), “La scopritore della illuminazione elettrica fu Edison” (“*The discoverer of electric illumination was Edison*”), where different synonyms of “inventore” (“*inventor*”) and “luce elettrica” (“*electric light*”) appear. In the experiment reported in section 3 Italian synonyms have been manually extracted from the ItalianWordnet database (Roventini et al., 2000), a further extension of the Italian Wordnet produced by the EuroWordNet project (Vossen, 1998). Once the correct synset for a basic keyword is selected, its synonyms are added to the expansion list. In the near future we plan to automate the process of synset selection using word domain disambiguation, a variant of word sense disambiguation based on subject field code information added to WordNet (Magnini and Cavaglià, 2000).

1.4 Expansion chains

The expansions described in the previous sections could be recursively applied to every lemma derived by a morphological or a synonym expansion. For example, at the first expansion level we can pass from “inventore”

“*inventor*” to its synonym “scopritore” “*discoverer*”, from which in turn we can morphologically derive the noun “discovery”, and so on (cfr. Figure 2). This would allow the retrieval of answers such as “La scoperta della lampada ad incandescenza è dovuta a Edison” (“*The discovery of the incandescent lamp is due to Edison*”).

Although in the experiment reported in this paper we do not use recursive expansions (i.e. we stop at the first level of the expansion chain), a long term goal of this work is to verify their effects on the document relevance.

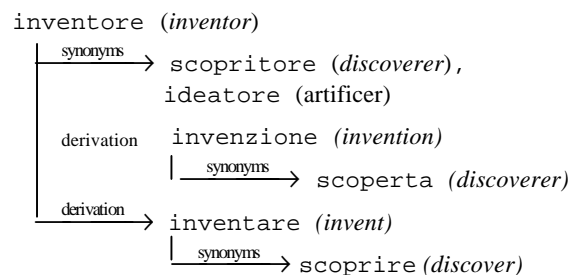


Figure 2: Lexical chain for “inventore” (“*inventor*”)

2 Query compositions

We wanted to take advantage of the “advanced” capabilities of the search engine. In particular we experimented the “Boolean phrase” modality, which allows the user to submit queries with keywords composed by means of logical operators. However we quickly realised that realistic choices were restricted to disjoint compositions of short AND clauses (i.e. with a limited number of elements, typically not more than four). This constrained us to two hypothesis, described in sections 2.2 and 2.3, which have been compared with a baseline composition strategy, described in 2.1.

2.1 Keyword “AND” composition search (KAS)

This search strategy corresponds to the default method that most search engines implement. Given a list of basic keywords, no expansion is performed and keywords are composed in an AND clause. An example is reported in Figure 3.

```

NL-QUESTION: Chi ha inventato la luce
elettrica? (Who invented the electric light?)
BASIC-KEYWORDS: inventore (inventor)
luce_elettrica (electric_light)
EXPANSIONS:
COMPOSITION: (inventore AND
luce_elettrica)

```

Figure 3: Example of AND composition search

2.2 Keyword expansion insertion search (KIS)

In this composition modality a disjunctive expression is constructed where each disjoint element is an AND clause formed by the base keywords plus the insertion of a single expansion. In addition, to guarantee that at least the same documents of the KAS modality are retrieved, both an AND clause with the basic keywords and all the single basic keywords are added as disjoint elements. Figure 4 reports an example. If the AND combination of the basic keywords produces a non empty set of documents, then the KIS modality should return the same set of documents rearranged by the presence of the keyword expansions. What we expect is an improvement in the position of a significant document, which is relevant when huge amounts of documents are retrieved.

```

NL-QUESTION: Chi ha inventato la luce
elettrica? (Who invented the electric light?)
BASIC-KEYWORDS: inventore (inventor)
luce_elettrica (electric_light)
EXPANSIONS:
inventore
  | synonyms → scopritore, ideatore
  | derivation → invenzione
  |               | synonyms → scoperta
  |               |
  | derivation → inventare
  |               | synonyms → scoprire
luce_elettrica
  | synonyms → lampada_a_incandescenza
COMPOSITION:
(OR (inventore AND luce_elettrica AND
scopritore)
OR (inventore AND luce_elettrica AND
ideatore)
OR (inventore AND luce_elettrica AND
invenzione)

```

```

OR (inventore AND luce_elettrica AND
scoperta)
OR (inventore AND luce_elettrica AND
inventare)
OR (inventore AND luce_elettrica AND
scoprire)
OR (inventore AND luce_elettrica AND
lampada_a_incandescenza)
OR (inventore AND luce_elettrica)
OR inventore OR luce_elettrica)

```

Figure 4: Example of expansion insertion composition

2.3 Keyword Cartesian composition search (KCS)

In this composition modality a disjunctive expression is constructed where each disjoint element is an AND clause formed by one of the possible tuple derived by the expansion set of each base keyword. In addition, to guarantee that at least the same documents of the KAS modality are retrieved, the single basic keywords are added as disjoint elements. Figure 5 reports an example.

As in the previous case we expect that at least the same results of the KAS search are returned, because the AND composition of the basic keywords is guaranteed. We also expect a possible improvement of the recall, because new AND clauses are inserted.

```

NL-QUESTION: Chi ha inventato la luce
elettrica?
BASIC-KEYWORDS: inventore
luce_elettrica
EXPANSIONS:
inventore
  | synonyms → scopritore, ideatore
  | derivation → invenzione
  |               | synonyms → scoperta
  |               |
  | derivation → inventare
  |               | synonyms → scoprire
luce_elettrica
  | synonyms → lampada_a_incandescenza
COMPOSITION:
(OR (inventore AND luce_elettrica)
OR (inventore AND lampada_a_incandescenza)
OR (scopritore AND luce_elettrica)
OR (scopritore AND lampada_a_incandescenza)
OR (ideatore AND luce_elettrica)
OR (ideatore AND lampada_a_incandescenza)
OR (invenzione AND luce_elettrica)
OR (invenzione AND lampada_a_incandescenza)
OR (scoperta AND luce_elettrica)
OR (scoperta AND lampada_a_incandescenza)

```

```

OR (inventare AND luce_elettrica)
OR (inventare AND lampada_a_incandescenza)
OR (scoprire AND luce_elettrica)
OR (scoprire AND lampada_a_incandescenza)
OR inventore OR luce_elettrica)

```

Figure 5: Example of Cartesian composition search

3 Comparison experiment

This section reports about the problems we faced with comparing the three search strategies presented in section 2. The question set, the document assessment and the scoring used in the experiment are described.

3.1 Creating the Question Set

Initially, a question set of 40 fact-based, short-answer questions such as “Chi è l’autore della Divina Commedia?” (“*Who is the author of The Divine Comedy?*”) was created. Language was Italian and each question was guaranteed to have at least one web document that answered the question. Ambiguous questions (about 15%) were not eliminated (see Voorhees, 2000 for a discussion). A total of 20 questions from the initial question set have been randomly selected, this way preventing possible bias in favour of queries that would perform better with lexical expansions. Figure 6 reports the final question set of the experiment.

1	Chi ha inventato la luce elettrica? (<i>Who invented the electric light?</i>)
2	Come si chiama l'autore del libro "I Malavoglia"? (<i>Who is the author of the book "I Malavoglia"?</i>)
3	Chi ha scoperto la legge di gravità? (<i>Who discovered the gravitational law</i>)
4	Chi ha inventato la stampa? (<i>Who is the inventor of printing</i>)
5	Chi ha vinto il campionato di calcio nel 1985 ? (<i>Who won the soccer championship in 1985?</i>)
6	Chi è il regista di "I Mostri" (<i>Who is the director of "I Mostri"</i>)
7	Quale attore ha recitato con Benigni nel film "Il piccolo Diavolo"? (<i>Who played with Benigni in the film "Il piccolo Diavolo"?</i>)
8	Chi ha ucciso John Kennedy? (<i>Who assassinated John Kennedy?</i>)
9	Chi detiene il recod italiano dei 200 metri? (<i>Who holds the Italian record for the 200-</i>

	<i>meters dash?</i>)
10	Chi è stato il primo uomo sulla Luna? (<i>Who was the first man on the moon?</i>)
11	Chi ha inventato il Lisp? (<i>Who is the inventor of the Lisp</i>)
12	Premio nobel per la letteratura nel 1998 (<i>1998 Nobel Prize in literature</i>)
13	Quale è il fiume più lungo del mondo? (<i>Which is the longest river of the world?</i>)
14	In quale squadra di calcio Italiana ha giocato Van Basten? (<i>Which Italian soccer team did Van Basten play in?</i>)
15	Chi ha vinto i mondiali di Calcio nel 1986? (<i>Who won the World Cup Soccer in 1986?</i>)
16	Chi ha progettato la Reggia di Caserta? (<i>Who was the architect of the Caserta royal palace?</i>)
17	Dove è nato Alessandro Manzoni? (<i>Where was Alessandro Manzoni born?</i>)
18	Quale è il lago più grande d'Italia? (<i>Which is the largest Italian lake?</i>)
19	Chi ha fondato la Microsoft? (<i>Who is the founder of Microsoft?</i>)
20	Chi è il padre della relatività? (<i>Who is the father of the relativity theory?</i>)

Figure 6: Question set used in the experiments.

Each question was then associated with a corresponding human-generated set of basic keywords, resulting in an ordered list of [nl-question, basic-keywords] pairs. We supposed a maximum of 3 basic keywords for each question, obtaining an average of 2.25. This is in line with (Jansen et al., 1998) where it is reported that, over a sample of 51.473 queries submitted to a major search service (Excite), the average query length was 2.35. Basic keywords are then expanded with their morphological derivations and synonyms (see Section 2), with an average of two expansions for question (min=0, max=6).

3.2 Document assessment

An automatic query generator has been realised that, given a question with its basic keywords and lexical expansions, builds up three queries, corresponding to KAS, KIS and KCS, and submits them to the search engine. Results are collected considering up to ten documents for search; then the union set is used for the evaluation experiment. There was no way for the assessor to relate a document to the search modality the document was retrieved by. Query

generation, web querying and result displaying were all been made runtime, during the evaluation session.

Fifteen researchers at ITC-irst were selected as assessors in the experiment. They were asked to judge the web documents returned by the query generator with respect to a given question, choosing a value among the following five:

- 1) *answer_in_context*: The answer corresponding to the question is recovered and the document context is appropriate. For example, if the question is “Who is the inventor of the electric light?” then “Edison” is reported in the document, in some way, as the inventor of the electric light and the whole document deals with inventors and/or Edison’s life.
- 2) *answer_no_context*: The answer to the question is recovered but the document context is not appropriate. (e.g. the document does not deal neither with inventors or Edison’s life).
- 3) *no_answer_in_context*: The answer corresponding to the question is not recovered but the document context is appropriate.
- 4) *no_answer_no_context*: The answer corresponding to the question is not recovered and the document context is not appropriate.
- 5) *no_document*: the requested document is not retrieved.

The following instructions were provided to assessors:

- The judgement has to be based on the document text only, that is no further links exploration is allowed.
- If a question is considered ambiguous then give it just one interpretation and use that interpretation to judge all question-related documents consistently. For example, if the question “Chi è il vincitore del Tour de France?” (“Who is the winner of the Tour de France?”) is considered ambiguous because the answer may change over time, then the assessor could decide that the correct interpretation is “Who is the winner of the 1999 Tour de France?” and judge all the documents consistently.
- A document contains the answer only if it is explicitly reported in the text. That is, if the question is “Who is the author of *Options*?” it is not sufficient that the string “Robert Sheckley” or “Sheckley” is in the text, but the document has to say that *Robert Sheckley* is the author of *Options*.

Each question was judged independently by three assessors. The number of texts to be judged for a question ranged from 10 to 18, with an average of 12. For each question k we obtained three sets $V_{KAS,k}$, $V_{KIS,k}$ and $V_{KCS,k}$ of $(pos, assessment)$ pairs corresponding to the three search methods, where pos is the position of the document in the ordered list returned by the search method, and $assessment$ is the assessment of one participant.

3.3 Assessment scoring

We eliminated all the $(pos, assessment)$ pairs whose $assessment$ was equal to *no_document*. Said i a $(pos, assessment)$ pair belonging to $V_{KAS,k}$, $V_{KIS,k}$ or $V_{KCS,k}$ we define:

$$r(i) = \begin{cases} 0 & \text{if } assessment \text{ is } no_answer_no_context \\ 1 & \text{if } assessment \text{ is } no_answer_in_context \\ 2 & \text{if } assessment \text{ is } answer_no_context \\ 3 & \text{if } assessment \text{ is } answer_in_context \end{cases}$$

Given a question k and a set V_k of $(pos, assessment)$ pairs corresponding to an ordered list L_k of documents, to evaluate the relevance of L_k with respect to k we have defined two relevance functions, defined in [1]: f_+ that considers the document position, and f_- that does not.

$$f_-(k) = \frac{\sum_{i \in V_k} v(i)}{m} \quad f_+(k) = \frac{\sum_{i \in V_k} v(i) / p(i)}{\sum_{j=1}^m 1/j}$$

where

- $p(i)$ is the position of the web document in the ordered list.

- $v(i) = \alpha(r(i)) \cdot r(i) + \beta(r(i))$

$\alpha(x), \beta(x) : \{0,1,2,3\} \rightarrow (0,1)$ are *tuning functions* that allow to weight the assessments.

- m is the maximum length of an ordered list of web documents.

For each search method we obtained a set of 20 (f_-, f_+) pairs by the assessing process, i.e., we obtained 20 $(f_-, f_+)_{KAS,k}$ pairs, 20 $(f_-, f_+)_{KIS,k}$ pairs and 20 $(f_-, f_+)_{KCS,k}$ pairs.

4 Results and discussion

During the assessing process, some requested URLs were not retrieved. We have a total of 546 URLs and 516 retrieved web documents, meaning that about 6% of URLs were not retrieved (see Table 1).

	KAS	KIS	KCS	Total
Total URLs	146	200	200	546
Retrieved URLs	137	191	188	516
% Retrieved URLs	94%	95%	94%	94%

Table 1: URLs returned by KAS, KIS and KCS methods and URLs retrieved during the assessing process.

Table 2 shows the assessments on the KAS search method, which we consider the baseline of the experiment, being search by keywords a standard search method on the Web.

Results are presented for three partitions of the question set. QS1 is the subset of questions whose number of morphological derivations and synonyms is higher than three; QS2 is the subset whose number of lexical expansions is equal to two or three; QS3 is the subset whose number of lexical expansions is lower than two. The table reports the average values of f (i.e. document order not considered) and f_+ (i.e. order considered) with respect to each partition. The obtained values, f 0.23 and f_+ 0.25, indicate that, on average, about 2 web documents have an *answer_in_context* assessment and 7 web documents have *no_answer_no_context* assessment out of 10 documents returned by this method.

	KAS			
	Mean		Sdev	
	f (- pos.)	f_+ (+pos.)	f (- pos.)	f_+ (+ pos.)
QS1	0.14	0.20	0.20	0.23
QS2	0.37	0.31	0.43	0.34
QS3	0.22	0.23	0.20	0.21
all	0.21	0.23	0.25	0.23

Table 2: Mean and standard deviation of the relevance values f (without position) and f_+ (with position) of retrieved web documents returned by KAS method.

Table 3 reports the relevance values for the documents retrieved respectively by KIS and KCS. For KIS we have a growth of the 19% and 13% compared with the KAS method. For KCS the average growth is 33 % and 22% compared with KAS. On QS2 there is a remarkable improvement in the KCS performances compared with KAS (+59% and +77%). In this case the average value of f_+ is greater than f , meaning that KCS recovers good web documents in a better position than KAS. On QS3 there is also a good performance of both KIS and KCS compared with KAS (+18% and +17% for KIS, +23% and +17% for KCS). On the contrary, on the subset QS1 both KIS and KCS performances are comparable to KAS.

	KIS		KCS	
	% KAS		% KAS	
	f (- pos.)	f_+ (+ pos.)	f (- pos.)	f_+ (+ pos.)
QS1	+7 %	-15%	+7 %	- 15 %
QS2	-3 %	+19 %	+59 %	+77 %
QS3	+18 %	+17 %	+23 %	+17 %
all	+19 %	+13 %	+33 %	+22 %

Table 3: KIS and KCS increasing of the average relevance with respect to KAS.

From the data presented here it does not emerge a clear correlation between the performance of a search method and the number of lexical expansions. It can be noted that both KIS and KCS perform quite well, compared with KAS, on the set of questions having no expansions. This can be explained because KIS and KCS create queries less restrictive than KAS and are able to recover the same documents of KAS as well as other documents that can be meaningful.

In case lexical expansions are present, the best performance compared with KAS is carried out by KCS method on question 1 (Figure 6), which have a total of four derivations and four synonyms. In this case KAS recovered two documents and KCS more than ten documents, improving also the *answer_in_context* assessments thanks to both the morphological derivation "invenzione" ("*invention*") and the synonym "lampadina_elettrica" ("*electric_lamp*").

It is not clear if synonyms affect search performance more than morphological derivation or vice versa. It seems that synonyms and morphological derivations are significant expansions in the same way. If we consider the set of the questions characterised by an improvement in the KCS and KIS performance compared with KAS performance, then there are four questions having the number of synonyms greater than the number of morphological derivations, three questions having the number of synonyms lower than the number of morphological derivations and three questions having the number of synonyms equal to the number of morphological derivations (zero included). If we consider the set of questions having the number of synonyms higher than the number of morphological derivations, then there are four cases out of eight where KIS and KCS enhance the performance of KAS. If instead we consider the set of questions having the number of synonyms lower than the number of morphological derivations there are three cases out of six where KIS and KCS enhance the performance of KAS.

Finally, Table 4 synthetically shows how KIS and KCS perform with respect to document “context retrieval”, that is the degree of contextual information provided by the document with respect to the question, no matter if the answer to the question was present or not in the document itself. To focus on context we set the tuning functions $\alpha(x)$ and $\beta(x)$ to $\alpha(0)=0$, $\alpha(1)=1$, $\alpha(2)=0$, $\alpha(3)=1/3$ and $\beta(x)=0$. The reason for considering a context retrieval score is that, in case the answer is not present, context increases the probability that other relevant documents can be found following hypertextual links, possibly including the correct answer to the question.

Results obtained with KIS and KCS confirm that they provide a significant increase (from 31% to 41%) of context retrieval score.

	% context retrieval increasing with respect to KAS	
	$f_- (- pos.)$	$f_+ (+ pos.)$
KIS	37 %	+ 31 %
KCS	41 %	+ 38 %

Table 4: KIS and KCS context retrieval increasing with respect to KAS.

Conclusion

A comparative experiment among three search strategies has been carried out with the aim of estimating the benefits of lexical expansions and of composition strategies over the basic keywords of a query. Results lead us believe that search strategies that combine a number of linguistic optimizations with a proper Boolean composition can improve the performance of an existing search engine on the web. In particular given KAS (no expansions, with AND composition search) as baseline, KIS (expansion insertion search) performs better but one case (i.e. with expansions greater than 3) and KCS (Cartesian composition search) performs better than KIS. Furthermore, KCS has a maximum performance, with expansions equal to 2 or 3, significantly higher than KIS, probably because KCS retrieves web documents that are not retrieved by KIS, which basically rearranges the order of KAS documents.

At present we still have no clear data to determine which number and which kind (i.e. morphological derivations and synonyms) of lexical expansions performs better for a single question, even if all the three search strategies definitely perform better with questions with a limited number of expansions (i.e. two or three). An evaluation that will take into considerations such variations is planned for the near future. A crucial related problem for the future is that of the automatic evaluation of the search strategies (see Breck et al., 2000), which will enormously speed up the design and evaluation cycle.

The experiments reported in this paper are part of a feasibility study for the realisation of a Natural Language Based search engine on the Web. At the present state of development, some steps in the query expansion (i.e. multiword recognition and synset selection) have been done manually, while both the keyword composition and the actual search are automatic and very efficient. In order to completely automate the process, the main source of inefficiency is likely to be keywords disambiguation in WordNet. The idea is to use a two stage disambiguation algorithm (Voorhees, 1998), based on topic information, which

performs linearly with respect to the number of words to be disambiguated.

References

- Breck, E.J., Burger J.D., Ferro L., Hirschman L., House D., Light M., Mani I (2000) *How to Evaluate Your Question Answering System Every Day ...and Still Get Real Work Done*. Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pp. 1495-1500.
- Disc (1997) *Dizionario Italiano Sabatini Coletti*, Firenze, Giunti.
- Gonzalo J., Verdejo F., Peters C. and Calzolari N. (1998) *Applying EuroWordnet to Cross-Language Text Retrieval*. Computers and the Humanities, 32, 2-3, pp. 185-207 .
- Gonzalo J., Verdejo F., Chugur I., Cigar J. (1998) *Indexing with WordNet synsets can improve text retrieval*. Proceedings of the Workshop "Usage of Wordnet in NLP systems" Coling-ACL.
- Jansen B. J., Spink A., Bateman J., Saracevic T. (1998) *Real life information retrieval: A study of user queries on the Web*. SIGIR Forum, 32(1), 5-17.
- Magnini B. and Cavaglià G. (2000) *Integrating Subject Field Codes into Wordnet*. Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pp. 1413-1418.
- Mandala R., Takenobu T. and Hozumi T. (1998) *The Use of WordNet in Information Retrieval*. Proceedings of Coling-ACL.
- Roventini A., Alonge A., Bertagna F., Calzolari N., Magnini B., Martinelli R. (2000) *ItalWordNet, a large semantic database for Italian*. Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pp. 783-790.
- Verdejo F., Gonzalo J., Penas A., Lopez F., Fernandez D. (2000) *Evaluating wordnets in Cross-Language Information Retrieval: the ITEM search engine*. Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pp. 1769-1774.
- Voorhees, Ellen M., Using WordNet for Text Retrieval, in Fellbaum C. (1998) *WordNet, an Electronic Lexical Database*. MIT Press.
- Voorhees, Ellen M. and Tice Dawn M. (2000) *Implementing a Question Answering Evaluation*. Proceedings of the Workshop "Using Evaluation within HLT Programs: Results and Trends", Athens, Greece, May 30, 2000.
- Vossen P. (1998) *EuroWordnet: a multilingual database with lexical semantic networks*. Kluwer Academic Publishers .