Terminological variants for document selection and question/answer matching

Olivier Ferret **Brigitte Grau** **Martine Hurault-Plantet**
Gabriel Illouz **Christian Jacquemin**

LIMSI-CNRS
Bat.508 Université ParisXI
91403 Orsay, France
{ferret, grau, mhp, gabrieli, jacquemin}@limsi.fr

Abstract

Answering precise questions requires applying Natural Language techniques in order to locate the answers inside retrieved documents. The QALC system, presented in this paper, participated to the Question Answering track of the TREC8 and TREC9 evaluations. QALC exploits an analysis of documents based on the search for multi-word terms and their variations. These indexes are used to select a minimal number of documents to be processed and to give indices when comparing question and sentence representations. This comparison also takes advantage of a question analysis module and recognition of numeric and named entities in the documents.

1 Introduction

The Question Answering (QA) track at TREC8 and TREC9 is due to the recent need for more sophisticated paradigms in Information Retrieval (IR). Question answering generally refers to encyclopedic or factual questions that require concise answers. But current IR techniques do not yet enable a system to give precise answers to precise questions. Question answering is thus an area of IR that calls for Natural Language Processing (NLP) techniques that can provide rich linguistic features as output. Such NLP modules should be deeply integrated in search and matching components so that answer selection can be performed on such linguistic features and take advantage of them. In addition, IR and NLP techniques have to collaborate in the resulting system in order to cope with large-scale and broad coverage text databases while deriving benefit from added knowledge.

We developed a system for question answering, QALC, evaluated in the framework of the QA tracks at TREC8 and TREC9. The QALC system comprises NLP modules for multi-word term and named entity extraction with a specific concern for term conflation through variant recognition. Since named entity recognition has already been described extensively in other publications (Baluja 1999), we present the contribution of terminological variants to adding knowledge to our system.

The two main activities involving terminology in NLP are term acquisition and term recognition. Basically, terms can be viewed as a particular type of lexical data. Term variation may involve structural, morphological, and semantic transformations of single or multi-words terms (Fabre and Jacquemin, 2000).

In this paper, we describe how QALC uses high level indexes, made of terms and variants, to select among documents the most relevant ones with regard to a question, and then to match candidate answers with this question. In the selection process, the documents first retrieved by a search engine, are then postfiltered and ranked through a weighting scheme based on high level indexes, in order to

retain the top ranked ones. Similarly, all systems that participated in TREC9 have a search engine component that firstly selects a subset of the provided database of about one million documents. Since a search engine produces a ranked list of relevant documents, systems then have to define the highest number of documents to retain. Indeed, having too many documents leads to a question processing time that is too long, but conversely, having too few documents reduces the possibility of obtaining the correct answer. For reducing the amount of text to process, one approach consists of keeping one or more relevant text paragraphs from each document retrieved. Kwok et al (2000), for instance use an IR engine that retrieves the top 300 sub-documents of about 300-550 words and, on the other hand, the FALCON system (Harabagiu et all 2000) performs a paragraph retrieval stage after the application of a boolean retrieval engine. These systems work on the whole database and apply a bag-of-words technique to select passages whereas QALC first retains a large subset of documents, among which it then selects relevant documents by applying richer criteria based on the use of the linguistic structures of the words.

QALC indexes, used for document selection, are made of single and multi-word terms retrieved by a 2-step procedure: (1) automatic term extraction from questions through part-of-speech tagging and pattern matching and (2) automatic document indexing through term recognition and variant conflation. As a result, linguistic variation is explicitly addressed through the exploitation of word paradigms, contrarily to other approaches like the one taken in COPSY (Schwarz 1988) where an approximate matching technique between the query and the documents implicitly takes it into account. Finally, terms acquired at step (1) and indexes from step (2) are also used by the matching procedure between a question and the relevant document sentences.

In the next section, we describe the architecture of the QALC system. Then, we present the question processing for term extraction. We continue with the description of FASTR, a transformational shallow parser that recognizes and marks the extracted terms as well as their linguistic variants within the documents. The two following sections present the modules of the QALC system where terms and variants are used, namely the document selection and question/answer matching modules. Finally, we present the results obtained by the QALC system as well as an evaluation of the contribution of this NLP technique to the QA task through the use of the reference collections for the QA track. In conclusion, suggestions for more ambitious, but still realistic, developments using NLP are outlined.

2 System Overview

Natural Language Processing components in the QALC system (see Figure 1) enrich the selected documents with terminological indexes in order to go beyond reasoning about single words. Rich linguistic features are also used to deduce what a question is about.

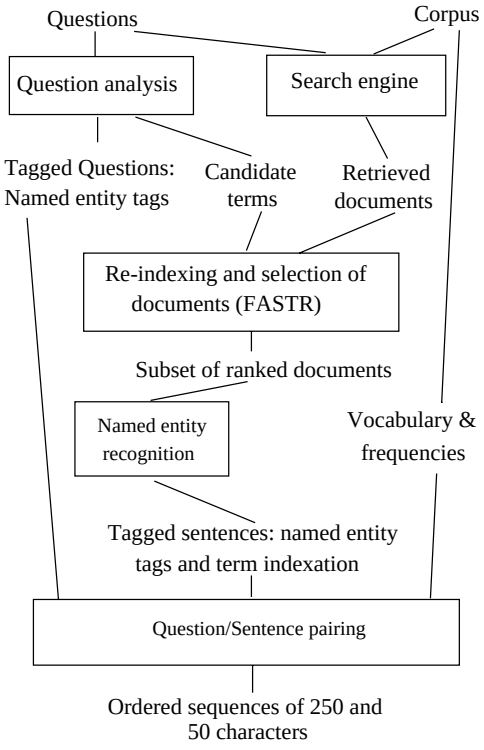


Figure 1. The QALC system

The analysis of a question relies on a shallow parser which spots discriminating patterns and assigns categories to the question. The categories correspond to the types of entities that are likely to constitute the answer to the question.

In order to select the best documents from the results given by the search engine and to

locate the answers inside them, we work with terms and their variants, i.e. morphologic, syntactic and semantic equivalent expressions. A term extractor has been developed, based on syntactic patterns which describe complex nominal phrases and their subparts. These terms are used by FASTR (Jacquemin 1999), a shallow transformational natural language analyzer that recognizes their occurrences and their variants. Each occurrence or variant constitutes an index that is subsequently used in the processes of document ranking and question/document matching.

Documents are ordered according to a weight computed thanks to the number and the quality of the terms and variants they contain. For example, original terms with proper names are considered more reliable than semantic variants. An analysis of the weight graph enables the system to select a relevant subpart of the documents, whose size varies along the questions. This selection takes all its importance when applying the last processes which consist of recognizing named-entities and analyzing each sentence to decide whether it is a possible answer or not. As such processes are time consuming we attempt to limit their application to a minimal number of documents.

Named entities are recognized in the documents and used to measure the similarity between the document sentences and a question. Named entities receive one of the following types: person, organization, location (city or place), number (a time expression or a number expression). They are defined in a way similar to the MUC task and recognized through a combination of lexico-syntactic patterns and significantly large lexical data.

Finally, the question/answer matching module uses all the data extracted from the questions and the documents by the preceding modules. We developed a similarity measure that attributes weights to each characteristic, i.e. named entity tags and terms and variants, and makes a combination of them. The QALC system proposes long and short answers. Concerning the short ones, the system focuses on parts of sentences that contain the expected named entity tags, when they are known, or on the largest subpart without any terms of the question.

3 Terms and Variants

3.1 Term extraction

For automatic acquisition of terms from questions, we use a simple technique of filtering through patterns of part-of-speech categories. No statistical ranking is possible because of the small size of the questions from which terms are extracted. First, questions are tagged with the help of the *TreeTagger* (Schmid 1999). Patterns of syntactic categories are then used to extract terms from the tagged questions. They are very close to those described by Justeson and Katz (1995), but we do not include post-posed prepositional phrases. The pattern used for extracting terms is:

(((((JJ | NN | NP | VBG)) ? (JJ | NN | NP | VBG) (NP | NN))) | (VBD) | (NN) | (NP) | (CD))

where NN are common nouns, NP proper nouns, JJ adjectives, VBG gerunds, VBD past participles and CD numeral determiners.

The longest string is acquired first and substrings can only be acquired if they do not begin at the same word as the superstring. For instance, from the sequence *name*NN *of*IN *the*DT *US*NP *helicopter*NN *pilot*NN *shot*VBD *down*RP, the following four terms are acquired: *US helicopter pilot*, *helicopter pilot*, *pilot*, and *shoot*.

The mode of acquisition chosen for terms amounts to considering only the substructures that correspond to an attachment of modifiers to the leftmost constituents (the closest one). For instance, the decomposition of *US helicopter pilot* into *helicopter pilot* and *pilot* is equivalent to extracting the subconstituents of the structure [*US* [*helicopter* [*pilot*]]].

3.2 Variant recognition through FASTR

The automatic indexing of documents is performed by FASTR (Jacquemin 1999), a transformational shallow parser for the recognition of term occurrences and variants. Terms are transformed into grammar rules and the single words building these terms are extracted and linked to their morphological and semantic families.

The *morphological family* of a single word *w* is the set *M*(*w*) of terms in the CELEX database

(CELEX 1998) which have the same root morpheme as *w*. For instance, the morphological family of the noun *maker* is made of the nouns *maker*, *make* and *remake*, and the verbs *to make* and *to remake*.

The *semantic family* of a single word *w* is the union $S(w)$ of the *synsets* of WordNet1.6 (Fellbaum 1998) to which *w* belongs. A synset is a set of words that are synonymous for at least one of their meanings. Thus, the semantic family of a word *w* is the set of the words *w'* such that *w'* is considered as a synonym of one of the meanings of *w*. The semantic family of *maker*, obtained from WordNet1.6, is composed of three nouns: *maker*, *manufacturer*, *shaper* and the semantic family of *car* is *car*, *auto*, *automobile*, *machine*, *motorcar*.

Variant patterns that rely on morphological and semantic families are generated through metarules. They are used to extract terms and variants from the document sentences in the TREC corpus. For instance, the following pattern, named NtoSemArg, extracts the occurrence *making many automobiles* as a variant of the term *car maker*:

VM('*maker*') RP? PREP? (ART (NN|NP)? PREP)? ART? (JJ | NN | NP | VBD | VBG)$^{[0-3]}$ NS('*car*')

where RP are particles, PREP prepositions, ART articles, and VBD, VBG verbs. VM('*maker*') is any verb in the morphological family of the noun *maker* and NS('*car*') is any noun in the semantic family of *car*.

Relying on the above morphological and semantic families, *auto maker*, *auto parts maker*, *car manufacturer*, *make autos*, and *making many automobiles* are extracted as correct variants of the original term *car maker* through the set of metarules used for the QA track experiment. Unfortunately, some incorrect variants are extracted as well, such as *make those cuts in auto* produced by the preceding metarule.

3.3 Document selection

The output of NLP-based indexing is a list of term occurrences composed of a document identifier *d*, a term identifier—a pair $t(q,i)$ composed of a question number *q* and a unique index *i*—, a text sequence, and a variation identifier *v* (a metarule). For instance, the following index :

LA092690-0038	t(131,1)
making many automobiles	NtoVSemArg

means that the occurrence *making many automobiles* from document *d*=LA092690-0038 is obtained as a variant of term *i*=1 in question *q*=131 (*car maker*) through the variation NtoVSemArg given in Section 3.2.

Each document *d* selected for a question *q* is associated with a weight. The weighting scheme relies on a measure of quality of the different families of variations described by Jacquemin (1999): non-variant occurrences are weighted 3.0, morphological and morpho-syntactic variants are weighted 2.0, and semantic and morpho-syntactico-semantic variants are weighted 1.0.

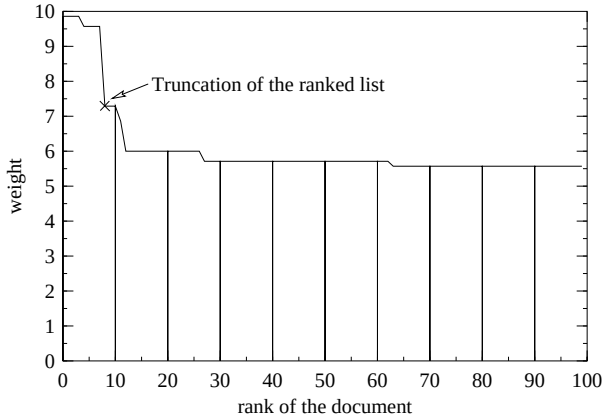
Since proper names are more reliable indices than common names, each term $t(q,i)$ receives a weight $P(t(q,i))$ between 0 and 1.0 corresponding to its proportion of proper names. For instance, *President Cleveland's wife* is weighted 2/3=0.66. Since another factor of reliability is the length of terms, a factor $|t(q,i)|$ in the weighting formula denotes the number of words in term $t(q,i)$. The weight $W_q(d)$ of a query *q* in a document *d* is given by the following formula (1). The products of the weightings of each term extracted by the indexer are summed over the indices $I(d)$ extracted from document *d* and normalized according to the number of terms $|T(q)|$ in query *q*.

$$W_q(d) = \sum_{(t(q,i),v)\in I(d)} \frac{w(v)\times(1+2P(t(q,i)))\times|t(q,i)|}{|T(q)|} \qquad (1)$$

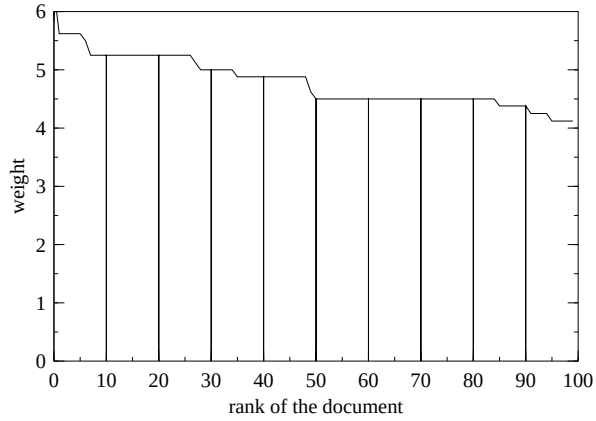
Mainly two types of weighting curves are observed for the retrieved documents: curves with a plateau and a sharp slope at a given threshold (Figure 2.a) and curves with a slightly decreasing weight (Figure 2.b).

The edge of a plateau is detected by examining simultaneously the relative decrease of the slope with respect to the preceding one, and the relative decrease of the value with respect to the preceding one. When a threshold is detected, we only select documents before this threshold, otherwise a fixed cutoff threshold is used. In our

experiments, for each query *q*, the 200 best ranked documents retrieved by the search engine[1] were subsequently processed by the re-indexing module. Our studies (Ferret et al. 2000) show that 200 is a minimum number such as almost all the relevant documents are kept. When no threshold was detected, we fixed the value of the threshold to 100.



(a) Question #87



(b) Question #86

Figure 2. Two types of weighting curve.

Through this method, the cutoff threshold is 8 for question #87 (*Who followed Willy Brandt as chancellor of the Federal Republic of Germany?*, Figure 2(a))[2] and 100 for question #86 (*Who won two gold medals in skiing in the Olympic Games in Calgary?*, Figure 2(b)). As indicated by Figure 2(a), there is an important difference of weight between documents #8 and #9. The weight of document #8 is 9.57 while the weight of document #9 is 7.29 because the term *Federal Republic* only exists in document #8. This term has a high weight because it is composed of two proper names.

[1] We used in particular Indexal (Loupy et al 1998), a search engine provided by Bertin Technologie.

[2] Questions come from the TREC8 data.

4 Question-Answer Matching

4.1 Question type categorization

Question type categorization is performed in order to assign features to questions and use these features for the similarity measurement between a question and potential answer sentences. Basically, question categorization allows the prediction of the kind(s) of answer, called target (for instance, NUMBER). Sentences inside the retrieved documents are labeled with the same tags as questions. During the similarity measurement, the more the question and a sentence share the same tags, the more they are considered as involved in a question-answer relation. For example:

Question:
How many people live in the Falklands?
—> target = NUMBER
Answer:
Falklands population of <bnumex TYPE=NUMBER> 2,100 <enumex> is concentrated.

We established 17 types of answer. Some systems define more categories. For instance Prager et all (2000) identify about 50 types of answer.

4.2 Answer Selection

In the QALC system, we have taken the sentence as a basic unit because it is large enough to contain the answer to questions about simple facts and to give a context that permits the user to judge if the suggested answer is actually correct. The module associates each question with the N_a most similar sentences (N_a is equal to 5 for the QA task at TREC).

The overall principle of the selection process is the following: each sentence from the documents selected for a question is compared with this question. To perform this comparison, sentences and questions are turned into vectors that contain three kinds of elements: content words, term identifiers and named entity tags. A specific weight (between 0 and 1.0) is associated

with each of these elements in order to express their relative importance.

The content words are the lemmatized forms of mainly adjectives, verbs and nouns such as they are given by the TreeTagger. Each content word in a vector is weighted according to its degree of specificity in relation to the corpus in which answers are searched through the tf.idf weighting scheme. For questions, the term identifiers refer to the terms extracted by the term extractor described in Section 3.1 and receive a fixed weight. In sentence vectors, term identifiers are associated with the normalized score from the ranking module (see Section 3.3). The named entity tags correspond to the possible types of answers, provided by the question analysis module. In each sentence these tags delimit the named entities that were recognized by the corresponding module of the QALC system and specify their type. Unlike term identifiers, named entity tags are given the same fixed weight in both sentence and question vectors because the matching module uses the types of the named entities and not their values.

In our experiments, the linguistic features (terms and named entities) are used to favor appropriate sentences when they have not enough content words in common with the question or when the question only contains a few content words. Thus, the weights of term identifiers or named entity tags are reduced by applying a coefficient in order to be globally lower than the weights of the content words.

Finally, the comparison between a sentence vector V_d and a question vector V_q is achieved by computing the following similarity measure:

$$sim(V_q, V_d) = \frac{\sum_i wd_i}{\sum_j wq_j} \qquad (2)$$

where wq_j is the weight of an element in the question vector and wd_i is the weight of an element in a sentence vector that is also in the question vector. This measure evaluates the proportion and the importance of the elements in the question vector that are found in the sentence vector with regards to all the elements of the question vector. Moreover, when the similarity value is nearly the same for two sentences, we favor the one in which the content words of the question are the least scattered.

The next part gives an example of the matching operations for the TREC8 question Q16 *What two US biochemists won the Nobel Prize in medicine in 1992?* This question is turned into the following vector:

two (1.0)	US (1.0)	biochemist (0.9)
nobel (1.0)	prize (0,6)	medicine (0,5)
win (0,3)	1992 (1.0)	<PERSON> (0.5)
16.01 (0.5)	16.04 (0.5)	

where <PERSON> is the expected type of the answer, 16.01 is the identifier of the *U S biochemist* term and 16.04 is the identifier of the *Nobel Prize* term.

The same kind of vector is built for the sentence *<NUMBER> Two </NUMBER> US biochemists, <PERSON> Edwin Krebs </PERSON> and <CITY> Edmond </CITY> Fischer, jointly won the <NUMBER> 1992 </NUMBER> Nobel Medicine Prize for work that could advance the search for an anti-cancer drug*, coming from the document FT924-14045 that was selected for the question Q16[3] :

two (1.0)	US (1.0)	biochemist (0.9)
nobel (1.0)	prize (0,6)	medicine (0,5)
win (0,3)	1992 (1.0)	Edwin (0.0)
Krebs (0.0)	Edmond (0.0)	Fischer (0.0)
work (0.0)	advance (0.0)	search (0.0)
anti-cancer (0.0)	jointly (0.0)	drug (0.0)
<PERSON> (0.5)	<NUMBER> (0.0)	<CITY>(0.0)
16.01 (0.5)	16.04 (0.3)	

where the weight 0.0 is given to the elements that are not part of the question vector. The term *US biochemist* is found with no variation and *Nobel Prize* appears as a syntactic variant. Finally, according to (2), the similarity measure between theses two vectors is equal to 0.974.

5 Results and Evaluation

We sent to TREC9 three runs whose variations concern the searched engine used and the length of the answer (250 or 50 characters). Among those runs, the best one obtained a score of 0.407 with 375 correct answers among 682 questions, for answers of 250 characters length. The score computed by NIST is the reciprocal mean of the rank, from 1 to 5, of the correct

[3] This sentence is taken from the output of the named entity recognizer.

answer. With this score, the QALC system was ranked 6th among 25 participants at TREC 9 QA task.

Document selection relies on a quantitative measure, i.e. the document weight, whose computation is based on syntactic and semantic indices, i.e. the terms and the terminological variants. Those indices allow the system to take into account words as well as group of words and their internal relations within the documents. Following examples, that we have got from selected documents for TREC9 QA task, show what kind of indices are added to the question words.

For the question 252 *When was the first flush toilet invented?* , one multi-word extracted term is *flush toilet*. This term is marked by FASTR when recognized in a document, but it is also marked when a variant is found, as for instance *low-flush toilet* in the following document sentence where *low-flush* is recognized as equivalent to *flush*:

> Santa Barbara , Calif. , is giving $ 80 to anyone who converts to a *low-flush toilet*.

252.01 flush toilet[JJ][NN]
low-flush[flush][JJ] toilet[toilet][NN]
1.00

In the given examples, after the identification number of the term, appears the reference term, made of the lemmatized form of the words and their syntactic category, followed by the variant found in the sentence, with each word, its lemmatized form and its category, and finally its weight.

In the example above, the term found in the sentence is equivalent to the reference term, and thus its weight is 1.00.

The second example shows a semantic variant. *Salary* and *average salary* are terms extracted from the question 337, *What's the average salary of a professional baseball player ?*. The semantic variant *pay*, got from WordNet, was recognized in the following sentence :

> Did the NBA union opt for the courtroom because its members, whose *average pay* tops $500000 a year, wouldn't stand still for a strike over free agency ?

337.01 salary[NN] pay[pay][NN] 0.25
337.00 average [JJ]salary[NN]
average[average][JJ] pay[pay][NN]
0.40

In order to evaluate the efficiency of the selection process, we proceeded to several measures. We apply our system on the material given for the TREC8 evaluation, one time with the selection process, and another time without this process. At each time, 200 documents were returned by the search engine for each of the 200 questions. When selection was applied, at most 100 documents were selected and subsequently processed by the matching module. Otherwise, the 200 documents were processed. The system was scored by 0.463 in the first case, and by 0.452 in the second case. These results show that the score increases when processing less documents above all because it is just the relevant documents that are selected.

The benefit from performing such a selection is also illustrated by the results given in Table 1, computed on the TREC9 results.

Number of documents selected by ranking	100	<<100
Distribution among the questions	342 (50%)	340 (50%)
Number of correct answers	175 (51%)	200 (59%)
Number of correct answer at rank 1	88 (50%)	128 (64%)

Table 1. Evaluation of the ranking process

We see that the selection process discards a lot of documents for 50% of the questions (340 questions are processed from less than 100 documents). The document set retrieved for those questions had a weighting curve with a sharp slope and a plateau as in Figure 2(a). QALC finds more often the correct answer and in a better position for these 340 questions than for the 342 remaining ones. The average number of documents selected, when there are less than 100, is 37. These results are very interesting when applying such time-consuming processes as named entity recognition and question/sentence matching. Document selection will also enable us to apply later on syntactic and semantic sentence analysis.

6 Conclusion

The goal of a question-answering system is to find an answer to a precise question, with a response time short enough to satisfy the user. As the answer is searched within a great amount of documents, it seems relevant to apply mainly numerical methods because they are fast. But, as we said in the introduction, precise answers cannot be obtained without adding NLP tools to IR techniques. In this paper, we proposed a question answering system which uses terminological variants first to reduce the number of documents to process while increasing the system performance, and then to improve the matching between a question and its potential answers. Furthermore, reducing the amount of text to process will afterwards allow us to apply more complex methods such as semantic analysis. Indeed, TREC organizers foresee a number of possible improvements for the future : real-time answering, evaluation and justification of the answer, completeness of the answer which could result from answers distributed along multiple documents, and finally interactive question answering so that the user could specify her/his intention. All those improvements require more data sources as well as advanced reasoning about pragmatic and semantic knowledge.

Thus, the improvements that we now want to bring to our system will essentially pertain to a semantic and pragmatic approach. For instance, WordNet that we already use to get the semantic variants of a word, will be exploited to refine our set of question types. We also plan to use a shallow syntactico-semantic parser in order to construct a semantic representation of both the potential answer and the question. This representation will allow QALC to select the answer not only from the terms and variants but also from the syntactic and semantic links that terms share with each other.

References

Baluja, S., Vibhu O. M., Sukthankar, R. 1999 Applying machine learning for high performance named-entity extraction. *Proceedings PACLING'99* Waterloo, CA. 365-378.

CELEX. 1998. http://www.ldc.upenn.edu/readme_files/celex.readme.html. Consortium for Lexical Resources, UPenns, Eds.

Fabre C., Jacquemin C, 2000. Boosting variant recognition with light semantics. *Proceedings COLING 2000*, pp. 264-270, Luxemburg.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA, MIT Press.

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Jacquemin C. (2000), QALC — the Question-Answering system of LIMSI-CNRS, pre-proceedings of TREC9, NIST, Gaithersburg, CA.

Harabagiu S., Pasca M., Maiorano J. 2000. Experiments with Open-Domain Textual Question Answering. *Proceedings of Coling'2000*, Saarbrucken, Germany.

Jacquemin C. 1999. Syntagmatic and paradigmatic representations of term variation. *Proceedings of ACL'99*. 341-348.

Justeson J., Katz S. 1995. Technical terminology: some linguistic properties and an algorithm for identification in texte. *Natural Language Engineering*. 1: 9-27.

Kwok K.L., Grunfeld L., Dinstl N., Chan M. 2000. TREC9 Cross Language, Web and Question-Answering Track experiments using PIRCS. *Pre-proceedings of TREC9*, Gaithersburg, MD, NIST Eds. 26-35.

Loupy C. , Bellot P., El-Bèze M., Marteau P.-F.. Query Expansion and Classification of Retrieved Documents, *TREC* (1998), 382-389.

Prager J., Brown, E., Radev, D., Czuba, K. (2000), One Search Engine or two for Question-Answering, NISTs, Eds., *Proceedings of TREC9*, Gaithersburg, MD. 250-254.

Schmid H. 1999. Improvments in Part-of-Speech Tagging with an Application To German. *Natural Language Processing Using Very Large Corpora*, Dordrecht, S. Armstrong, K. W. Chuch, P. Isabelle, E. Tzoukermann, D. Yarowski, Eds., Kluwer Academic Publisher.

Schwarz C. 1988. The TINA Project: text content analysis at the Corporate Research Laboratories at Siemens. *Proceedings of Intelligent Multimedia Information Retrieval Systems and Management (RIAO'88)* Cambridge, MA. 361-368.