QALC - the Question-Answering program of the Language and Cognition group at LIMSI-CNRS

Olivier Ferret Brigitte Grau Gabriel Illouz Christian Jacquemin
Nicolas Masson

LIMSI-CNRS
BP 133, 91403 ORSAY Cedex, FRANCE
{ferret,grau,gabrieli,jacquemin,masson}@limsi.fr

1 Introduction

In this report we describe the *QALC system* (the Question-Answering program of the Language and Cognition group at LIMSI-CNRS) which has been involved in the QA-track evaluation at TREC8. The purpose of the Question-Answering track is to find the answers to a set of 200 questions. The answers are text sequences extracted from the volumes 4 and 5 of the TREC collection. All the questions have at least one answer in the collection.

The basic architecture of *QALC* is composed of five parallel modules, two for the questions and three for the corpora, and a sixth pairing module which produces the sentences ranked by decreasing order of relevance. Figure 1 illustrates the main components of *QALC*. More details about each component will be given in the figures corresponding to the detailed descriptions of these subparts.

The *QALC* system relies mainly on genuine Natural Language Processing components. Most of the components rely on a tagged version of the corpus. We use the *TreeTagger* for this purpose (Stein and Schmid, 1995). The system is based on the following six modules:

Natural language question analysis The analysis of the questions relies on a shallow parser which spots discriminant patterns and assigns categories to the questions. The categories correspond to the types of entities which are likely to constitute the answer to this question.

Term extraction The term extractor is based on syntactic patterns which describe compound nouns. The maximal extension of these compounds is produced along with the plausible subphrases.

Automatic indexing & variant conflation Automatic indexing relies on *FASTR* (Jacquemin, 1999), a shallow transformational natural language analyzer which recognizes the occurrences and the variants of the terms produced by the preceding module. Each occurrence or variant constitutes an index to the document which is ultimately used in the process of document ranking and in the process of question/document pairing.

Named entity recognition Similarly, named entities are recognized in the documents in order to build indices which are used for measuring the degree of similarity between the questions and the document sentences. Named entities are extracted through lexico-syntactic patterns combined with significantly large lexical data.

Document ranking & thresholding The last two modules are information retrieval modules as opposed to the four preceding components. Documents are ranked according to a weighted measure of the indices produced by the automatic indexing and variant conflation module. Only the n best ranked documents are selected. A further selection of the documents is made if a plateau can be recognized in the relevance curve of the documents.

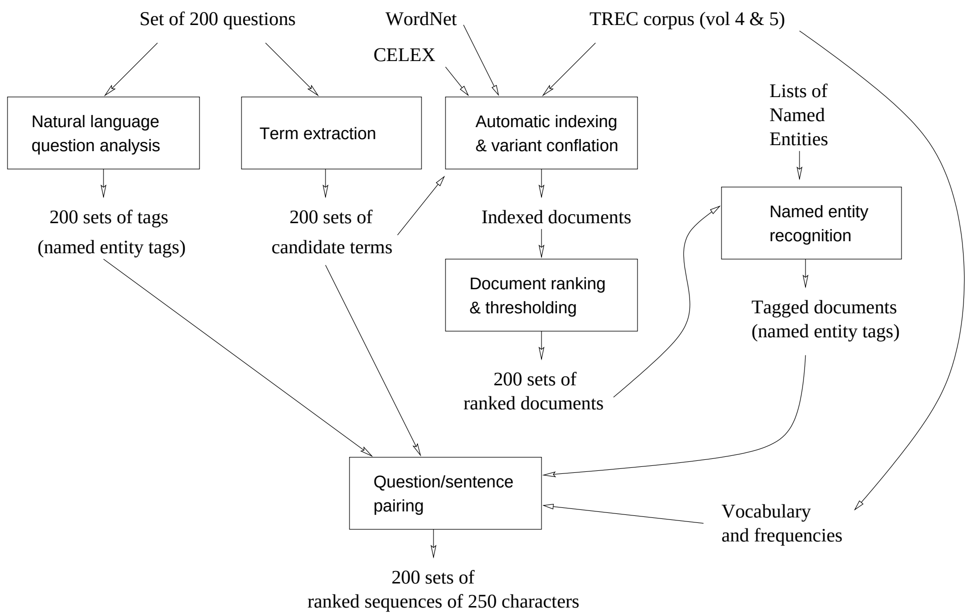


Figure 1: Flowchart of QALC.

Question/sentence pairing Finally, all the data extracted from the questions and the documents by the preceding modules is used by a pairing module to evaluate the degree of similarity between a document sentence and a question. The sentences are chosen from the documents selected by the preceding module.

The following sections present in detail the modules of the *QALC* system.

2 Natural Language Question Analysis

Question analysis is performed in order to assign features to questions and use these features for the pairing measurement between a query (question) and potential answer sentences (answer). Basically, question analysis allows the prediction of the kind(s) of answer, called *target* (for instance, ORGANIZATION). The retrieved documents (see Section 4.2) are processed in order to recognize the named entities. Named entities are labeled with the same set of tags as the questions. During the pairing measurement, the more the question and a sentence share the same tags, the more they are considered as involved in a question-answer relation.

Example:

Question: *How many people live in the Falklands?* → target = NUMBER

Answer: *... Falklands population of* <b_numex_TYPE=NUMBER> 2,100 <e_numex> *is concentrated.*

2.1 Target Set

The targets used are PERSON, ORGANIZATION, LOCATION (either CITY or PLACE), TIME-EXPRESSION (either DATE, TIME, AGE or PERIOD), and NUMBER (either LENGTH, VOLUME, DISTANCE, WEIGHT, PHYSICS or FINANCIAL). Some examples of sentences which can be associated with targets follow. Elements of sentences (called *triggers*) which are relevant to assign a given target are underlined.

PERSON: *Who was the first President of the USA?*

ORGANIZATION: *What laboratory discovered the AIDS virus?*

LOCATION:

PLACE: *What is the longest river in Asia? What is the name of the highest mountain in the world?*

TIME-EXPRESSION:

PERIOD: *During which period did the dinosaurs vanish?*

2.2 Types of Questions

Question analysis is performed by a specialized shallow parser. It is based on lexical, syntactic and semantic knowledge, i.e. some specific words, syntactic categories, grammar rules and semantic classes for nouns. We have identified six kinds of formulations for questions allowing the prediction of the kind of answer.

TYPE 1: the answer only depends on the interrogative pronoun. It is the case with *who/whom/whose*, *where* and *when*.

The following three patterns are dedicated to parse Which/What-questions. These patterns are identical whatever kind of answer is expected. Disambiguation is provided by the semantic class of the head noun belonging to the noun phrase (NPsem). These classes are detailed in section 2.3.

TYPE 2: *what/which ...be* NPsem

TYPE 3: *what/which* NPsem. A variant of this last rule is TYPE 3b, for questions about time. TYPE 3b: (PREP) *what/which* NPtime

TYPE 4: *what/which is the name of* NPsem.

The last two types entail the analysis of How-questions. TYPE 5 leads the system to choose a target according to the semantic category of the adjective, and, in TYPE 6 questions, the choice relies on the semantic category of the NP.

TYPE 5: *how* AdjSem.

TYPE 6: *how much/many* NPsem.

Our parser first tries to apply these rules on the beginning of the sentence. If none of them is fired, the parser tries to find one of these patterns inside the sentence, as in the sentence *The Faroes are part of what northern European country?* Some rules lead to find several targets in case of ambiguity. For example, the target of the question *Where is Bolivia?* may be either a LOCATION-STATE or a LOCATION-CITY.

2.3 Semantic Categories

Each semantic category corresponds to a target. The category of a noun phrase is based on the semantic class of its main noun. A noun phrase is made of a determiner (DET), followed by successive adjectives (JJ), nouns (N) with a possible possessive case (Poss), or verbs at the gerundive (V_{BG}) or past participle (V_{BD}) form. We have decided to search for the largest noun phrase structure:

$$\text{(DET) } (\text{JJ}|\text{N}_{\text{Poss}}|\text{N}|\text{V}_{\text{BG}}|\text{V}_{\text{BD}})^{\star} \text{ N} \tag{1}$$

The head of a noun phrase is the last noun of the longest match. For example, in *Johnny Mathis' high school track **coach***, *coach* is recognized as the head. We have established 17 semantic classes, hierarchically structured as shown in Figure 2.

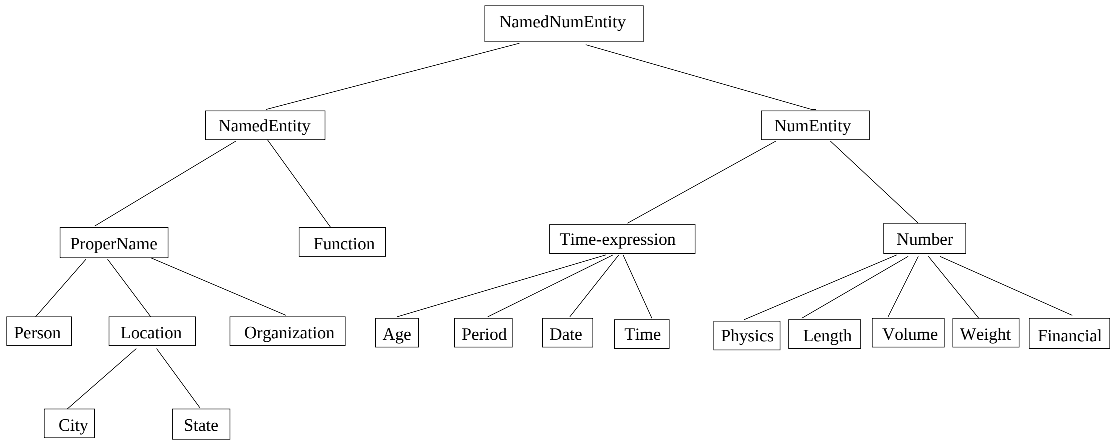


Figure 2: The semantic classes

Let us give some examples to illustrate our recognition of NPs. The NP is in italics, its head is underlined and the semantic category of the NP is given after the arrow:

What is *the number* of buffaloes → NUMBER
What *two companies* ... → ORGANIZATION
What *date/year* → DATE
Which *country* → STATE
What is *the world record time* → TIME
What is *the capital* of ... → CITY

Some adjectives used with *how*, as *tall*, *long*, etc., are also classified in semantic classes that belong to the sub-tree representing numeric entities.

3 Term Extraction

The relevant documents are retrieved through a pairing procedure which relies on a measure of similarity. The similarity between a document and a query, detailed in Section 6, is computed on the basis of common terms and on the correspondence between the tags assigned to the question and the named entities extracted from the sentences. Terms are extracted from the questions and sentences are indexed by these terms. In this section, we describe the acquisition of terms from the questions. In the next section, the process of indexing sentences with these terms is detailed.

As 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.

The questions are first tagged with the help of the *TreeTagger*. Then, patterns of syntactic categories are used to extract terms from the tagged corpora. They are very close to those in (Justeson and Katz, 1995), but we do not include post-posed prepositional phrases. The pattern used for extracting terms is[1]:

$$(((((\mathrm{JJ}|\mathrm{N_N}|\mathrm{N_P}|\mathrm{V_{BG}})\ )?(\mathrm{JJ}|\mathrm{N_N}|\mathrm{N_P}|\mathrm{V_{BG}})\ (\mathrm{N_P}|\mathrm{N_N})))|(\mathrm{V_{BD}})|(\mathrm{N_N})|(\mathrm{N_P})|(\mathrm{CD})) \tag{2}$$

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 $name_{\mathrm{N_N}}$ of_{IN} the_{DT} $US_{\mathrm{N_P}}$ $helicopter_{\mathrm{N_N}}$ $pilot_{\mathrm{N_N}}$ $shot_{\mathrm{V_{BD}}}$ $down_{\mathrm{RP}}$, 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

[1] $\mathrm{N_N}$ are common nouns, $\mathrm{N_P}$ proper nouns and CD numeral determiners.

of *US helicopter pilot* into *helicopter pilot* and *pilot* is equivalent to extracting the subconstituents of the structure [*US* [*helicopter* [*pilot*]]].

4 Automatic Indexing and Document Ranking

The selection of relevant documents relies on an NLP-based indexing composed of both single-word and phrase indices and linguistic links between the occurrences and the original terms. The original terms are those extracted from the questions and presented in Section 3. The tool used for extracting text sequences that correspond to occurrences or variants of these terms is *FASTR* (Jacquemin, 1999). The ranking of the documents relies on a weighted combination of the terms and variants extracted from the documents.

4.1 NLP-based Indexing through *FASTR*

The automatic indexing of documents is performed by *FASTR* , a transformational shallow parser for the recognition of term occurrences and variants. The terms acquired in Section 3 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 which 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. The following pattern[2] extracts the occurrence *making many automobiles* as a variant of the term *car maker*:

$$\mathrm{V_M}('maker')\ \mathrm{RP}^?\ \mathrm{PREP}^?\ (\mathrm{ART}\ (\mathrm{N_N}|\mathrm{N_P})^?\ \mathrm{PREP})^?\ \mathrm{ART}^?\ (\mathrm{JJ}\,|\,\mathrm{N_N}\,|\,\mathrm{N_P}\,|\,\mathrm{V_{BD}}\,|\,\mathrm{V_{BG}})^{0-3}\ \mathrm{N_S}('car') \tag{3}$$

$\mathrm{V_M}('maker')$ is any verb in the morphological family of the noun *maker* and $\mathrm{N_S}('car')$ 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 metarule set 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.

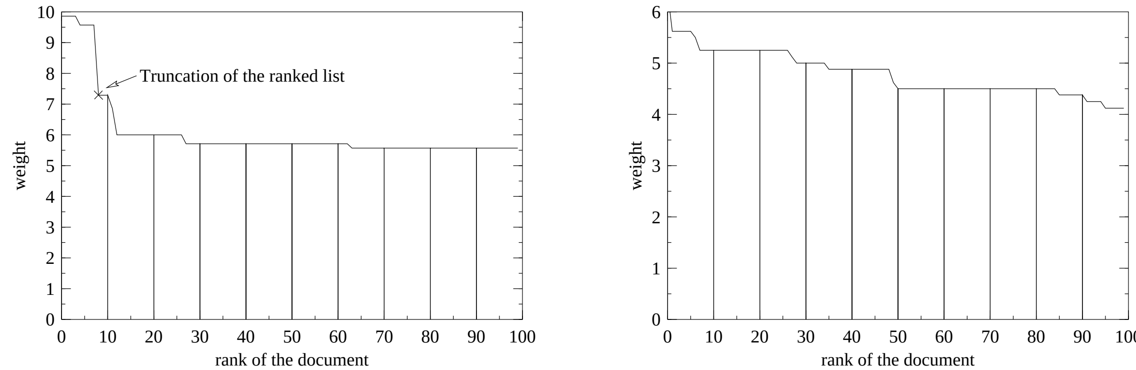
4.2 Document Ranking

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:

$$\text{LA092690-0038}\quad t(131,1)\quad \textit{making many automobiles}\quad \text{NtoVSemArg} \tag{4}$$

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 given in Section 4.1.

[2] RP are adverbs, PREP prepositions, ART articles, and V verbs.



(a) Question #87 **(b)** Question #86

Figure 3: Two types of weighting curve.

Each document d in the collection is associated with a vector $W(d) = (W_q(d))_{q \in \{1,\ldots,200\}}$ of 200 weights, one for each question q. The weighting scheme relies on a measure of quality of the different families of variations described in (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 $\frac{2}{3} = 0.66$. 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 (5). The products of the weightings of each term extracted by the indexer are summed over the indices $\mathcal{I}(d)$ extracted from document d and normalized according to the number of terms $|\mathcal{T}(q)|$ in query q.

$$W_q(d) = \sum_{(t(q,i),v) \in \mathcal{I}(d)} \frac{w(v) \times (1 + 2P(t(q,i))) \times |t(q,i)|}{|\mathcal{T}(q)|} \tag{5}$$

For each query q, the 100 best ranked documents are retrieved. 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 3.a) and curves with a slightly decreasing weight (Figure 3.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.

$$\text{if} \quad \frac{W_q(d_2)}{W_q(d_1)} \leq 0.5 \text{ then } i_0 = 2 \tag{6}$$

$$\text{else} \quad i_0 = \min\left\{ i \in \{3,\ldots,100\} \bullet \left(\frac{W_q(d_i) - W_q(d_{i-1})}{W_q(d_{i-1}) - W_q(d_{i-2})} \geq 0.5 \wedge \frac{W_q(d_i)}{W_q(d_{i-1})} \leq 0.8 \right) \right\} \cup \{100\} \tag{7}$$

Through this method, the threshold i_0 is 8 for question #87 (*Who followed Willy Brandt as chancellor of the Federal Republic of Germany?*, Figure 3.a) and 100 for question #86 (*Who won two gold medals in skiing in the Olympic Games in Calgary?*, Figure 3.b). As indicated by Figure 3.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 higher weight because it is composed of two proper names.

The i_0 best ranked documents are then processed by the question/sentence pairing module presented in Section 6. At the document level, this module relies on single words, term indices, and named entities. Term indices have been presented in this section, named entities are presented in the next one.

5 Named Entity Recognition

Named entities are recognized in the documents in order to build indices which are used for measuring the degree of similarity between the questions and the document sentences. Named entities receive one of the following types: `PERSON, ORGANIZATION, LOCATION, NUMBER`. They are defined in a similar way to the MUC task (Grishman and Sundheim, 1995) and recognized through a combination of

- lexical lookup (for syntactic or semantic tags on the single words) and rules which use these tags together with lexical elements; and
- dictionary lookup (the direct access to lists of named entities).

The three lists used for lexical lookup are CELEX (CELEX, 1998), a lexicon of 160,595 inflected words with associated lemma and syntactic category, a list of 8,070 first names (6,763 of which are from the CLR (CLR, 1998) archive at New Mexico State University) and a list of 211,587 family names from the CLR archive at New Mexico State University.

5.1 Numeric Entities

This category groups all kinds of number formulations, time expressions, and specialization of numbers according to their units, even if they are not expressed with numbers. Numeric entities are given a tag among FINANCIAL-AMOUNT, LENGTH, VOLUME, WEIGHT, PHYSICS, DATE, PERIOD, and NUMBER.

The recognition of numeric entities is performed in three steps:

- Firstly, we recognize basic entities as cardinal and ordinal numbers, either written with digits or with letters.
- In a second stage, we apply rules in order to recognize complex numeric entities such as monetary amounts, distances, or weights.
- In a third stage, we apply rules in order to recognize time expressions (labeled as TIMEX) such as dates, times, ages and periods. These rules use the output of the first set of rules dedicated to basic numeric entities such as cardinal and ordinal numbers (written in digits or letters).
- In a final stage, all basic numeric entities that have not been included into a complex entity are tagged as NUMBER.

5.2 Organizations, Persons and Locations

A list of 22,095 companies from the Wall Street Research Network and 649 organization obtained through lexical acquisition from the Internet is used to spot organization names. In addition, a set of rules recognizes organizations which are noun phrases that either begin with a specific modifier (such as *Christian*, *Democratic*, *Federal*...) or have a specific head word (such as *Academy*, *Administration*, *Association*...).

A word which belongs to the list of proper names exploited during the lexical lookup is tagged as a person. In addition, pairs of capitalized words that begin with a first name, or triples of capitalized words that begin with a title (such as *Dr*, *President*, *Ayatollah*...) are tagged as persons.

Simple anaphora which correspond to the elision of a subpart of a proper name are handled by the recognizer: a rule states that all the occurrences of a $\langle name \rangle$ inside a single document are references to the person identified by the rule $\langle firstName \rangle$ $\langle name \rangle$.

The location recognition module uses two lexical ressources from the CLR: a list of 7,813 city names and a list of 1,144 country names. No rules have been written for these entities so far.

6 Question/Sentence Pairing

This section presents the module that selects for each question a list of 5 ranked responses that are no longer than 250 characters each. This module relies on the results of all the preceding modules:

- each question is assigned a set of terms and one or several categories according to its focus;
- a set of documents is selected for each question. In each of them, named entities and terms extracted from the questions are tagged.

6.1 General Principles

The *Question/Sentence Pairing* module relies on two main choices, close to those that support the preliminary version of the Deep Read system (Hirschman et al., 1999). Firstly, we take the sentence unit as our basic unit for the answers to the questions, given that the $QALC$system aims at finding precise information that can be expressed in a single clause. Secondly, the *Question/Sentence Pairing* module directly searches for the possible answers of a question in all the documents selected by the *Document Ranking & Thresholding* module without trying to delimit smaller units. Such a choice allows the scanning of a large set of possible answers without too high a cost.

Considering the two preceding choices, the *Question/Sentence Pairing* module is based on a very simple principle: we compare each sentence from the selected documents for a question to this question and we always keep the five sentences that are the most similar to the question. This comparison is done first by transforming both the question and each document sentence into vectors and then, by computing a similarity measure between these two vectors. As is usual in the field of Information Retrieval, this similarity measure basically relies on the words of both the question and the sentences from the documents. But in our case, it also takes into account and merges all the linguistic information from the preceding modules. The final step of the module consists of cutting down the sentences that are longer than 250 characters. This is done here by simply removing the first and the last characters until we reach the fixed length.

6.2 Basic Similarity Measure

The similarity evaluation starts by turning sentences (questions and document sentences) into vectors of words. Such a vector only contains the most significant words of the primary sentence, i.e. mainly its content words: nouns and proper nouns, verbs, adjectives (including comparatives and superlatives), adverbs (including superlatives and comparatives), foreign words, symbols and cardinal numbers. These words are lemmatized and we take their canonical form as a reference in the vectors. We also have a short stop-list in order to remove some frequent words that are selected according to the previous criterion but that are not meaningful in our task. All this selection process relies on the morpho-syntactic tagging done by the *TreeTagger*, which also performs the lemmatizing.

Each word in a vector is weighted according to its importance in relation to the *Question/Answering* corpus. This importance is evaluated by using the *tf.idf* weighting policy, as it is often done in Information Retrieval. Word order in vectors is not significant because our similarity measure does not take this parameter into account. We think that it is too restrictive a constraint considering the kind of processing we do.

The similarity measure between a vector representing a question, V_q, and a vector representing a document sentence, V_d, is given by:

$$sim(V_q, V_d) = \frac{\sum_i wd_i}{\sum_j wq_j} \tag{8}$$

with wq_j, the weight of a word in the question vector and wd_i, the weight of a word in a sentence vector that is also in the question vector.

This measure evaluates the proportion and the importance of the words in the question vector that are present in the sentence vector with regard to all the words in the question vector. It is not symmetrical:

it favors the question point of view. It focuses on the similarities between the questions and the document sentences because firstly, there are too many differences at this level in comparison with the similarities, and then, these differences are globally not relevant.

On the other hand, we take into account the difference in length between a question and a document sentence. This criterion is used as a secondary key for sorting the sentences that are selected as possible answers to a question : if two sentences have the same similarity value, we sort as first the sentence that has the closest length with regard to the question.

6.3 Similarity Measure with Linguistic Features

We have chosen to take into account the results of the previous modules without changing our basic mechanism. Two kinds of linguistic features are considered: terms and named entities. Both of them are added to the vectors as if they were new significant words.

Terms Each of the terms extracted from a question has a unique identifier, which is used for marking the occurrences and the variants of this term both in the questions and in the document sentences. In the sentences, this identifier is associated with a score that reflects the distance between the found variant and its reference form in the question (see Section 4.2). In a question vector, we add the term identifier with a default weight of 0.5. In a sentence vector, we add the identifier of the recognized terms with a weight equal to the score of the variant divided by the maximum possible score.

Named entities Each recognized named entity is marked with a specific tag according to its type (see Section 5). On the other hand, the kind of the answer expected for each question is determined by the *Question Analysis* module. Thus, for a question, we add the tag(s) of the expected type(s) of the answer to its vector and for a sentence, we add the tags of the named entities that have been recognized in the sentence to its vector. In both cases, each tag is given a fixed weight, which is set to 0.5.

The weights associated with the term identifiers and with the named entity tags have been experimentally set. They are globally lower than the weights of the words. Thus, as for the difference of length, these linguistic features are used as a secondary criterion in the similarity measure. They help in increasing the rank of an answer which has already been selected, but the selection of the possible answers is mainly based on the number and the importance of the words shared by the question and these answers.

7 Results

Our official results are the following: our mean reciprocal rank is 0.341, with 88 questions answered. More precisely, we got 56 answers at rank 1, 12 at rank 2, 9 at rank 3, 7 at rank 4 and 4 at rank 5. Globally, this means that when an answer is found, it often appears on top position. Since our sentence truncating procedure for producing 250 character answers is very rough, our system sometimes missed the correct answer although it found a correct sentence. We think that applying very simple heuristics based on the recognized named entities could easily solve this problem.

One important aspect of the Question/Answering task is the ability to determine the expected type of answer (i.e. the target) in order to match it with named entities that are recognized in documents. Hence, we have analyzed our results according to this criterion. Among the 198 questions, a target was identified for 162 of them. The following table shows the percentage of correct answers in relation to the target type.

Target name	Correct answers (%)	Target name	Correct answers (%)
none	47	location	42
time-expression	27	number	65
person	49		

These results lead us to conclude that the recognition of a target does not influence the system's ability to find a correct answer, which seems rather contradictory with the main trends of the best participants. However our results for numbers, that are easily recognizable, comply with these trends and suggest that we have to enhance our named entity recognition module. We also think that our pairing module takes into account too weakly the concordance between the targets of questions and the named entities of documents.

8 Conclusion and Future Developments

Since the LIMSI laboratory did not have experience in the development of Question-Answering system before participating in the QA-track this year, many research issues have arisen from the construction of our system. Among the future developments that we are considering for our next participation in the QA-track are:

- exploring different weighting schemes for the computation of the similarity of a question and a sentence. For instance, when pairing a sentence with a question requiring a date, named entities denoting a date in sentences should be weighted higher;
- answer unit could be enlarged and position of indices inside a document could be accounted for in order to focus on the units that gather the largest number of indices and which are more likely to provide the answer;
- cascaded indexing with *FASTR* would optimize the computation by running *FASTR* on a smaller part of the collection;
- term acquisition could be improved through a disambiguation of long noun phrases and a better part-of-speech tagging of the questions;
- named entity recognition could be improved through machine learning techniques (Baluja, Mittal, and Sukthankar, 1999).

References

Baluja, Shumeet, Vibhu O. Mittal, and Rahul Sukthankar. 1999. Applying machine learning for high performance named-entity extraction. In *Proceedings, PACLING'99*, pages 365–378, Waterloo, CA.

CELEX. 1998. *www.ldc.upenn.edu/ readme_files/celex.readme.html*. Consortium for Lexical Resources, UPenn.

CLR. 1998. *http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat#D3*. Consortium for Lexical Resources, NMSU, New Mexico.

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

Grishman, R. and B. Sundheim. 1995. Design of the muc-6 evaluation. In NIST, editor, *the Sixth Message Understanding Conference (MUC-6)*, Columbia, MD. NIST, Morgan-Kauffmann Publisher.

Hirschman, L., M. Light, E. Breck, and J. D. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings, ACL'99*, pages 325–332, University of Maryland.

Jacquemin, Christian. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings, ACL'99*, pages 341–348, University of Maryland.

Justeson, John S. and Slava M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Stein, Achim and Helmut Schmid. 1995. Etiquetage morphologique de textes français avec un arbre de décision. *t.a.l.*, 36(1-2):23–36.