Document selection refinement based on linguistic features for QALC, a Question Answering system

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

LIMSI-CNRS
BP 133
91403 Orsay cedex France
{ferret, bg, mhp, gabrieli, jacquemin}@limsi.fr

ABSTRACT
Answering to 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. This selection is fully significant when applying further time consuming modules.

1 INTRODUCTION
A question answering system has to find a precise answer in response to a natural language question. Question answering (QA) issue was introduced at the end of the seventies by Lehnert (1977). She mainly focused on question taxonomy providing question type classes that still now inspire many question answering systems. Her QUALM system used complex semantic information but retrieved answers in a small size corpus. With the expansion of Internet, the QA issue gained a fresh interest both from Information Retrieval (IR) and Natural Language Processing (NLP) communities. But the retrieval documents corpora thereby become very large. For instance, the natural language question answering system START (Katz, 1997) is connected to the World Wide Web since 1993. Questions are requests for information from START’s knowledge base. Indeed, START attempt to solve the problem of retrieving information in a large and unstructured collection of documents by annotating documents. So, START’s knowledge base is made of natural language descriptive information attached to the documents available through the Web. As a token of interest in this issue, the TREC (Text REtrieval Conference) evaluation conference added two years ago the Question Answering track to the other IR tasks yet existing. Participants to TREC are provided with a very large corpus of documents (about 3 gigabytes) from which they have to extract answers in response to a list of questions.

Answering precise questions requires applying NLP techniques in order to locate the answers inside the documents of the corpus. As such techniques are time consuming, the size of the corpus to process is of great importance. To solve this problem, all systems presented at TREC used a search engine in order to select firstly a subset of relevant documents from the corpus, and then to apply NLP techniques on this subset. Most of the systems processed document selection either by getting few of the best ranked documents retrieved by the search engine (Litkowski, 2000 ; Vicedo and Ferrandez, 2000), or by selecting relevant paragraphs within a larger amount of the best ranked documents (Harabagiu et al. 2000, Kwok et al. 2000). The first solution generally leads to lower results when selecting too few documents because many documents containing the correct answer are then eliminated. The solution that we used in QALC, our question answering system, consists in using a NLP based re-indexing of the best ranked documents retrieved by the search engine and a subsequent re-ranking of the documents. The new index is made of the multi-word terms extracted from the question and their morphological and syntactic variants retrieved in the documents. The ranking of the documents relies on a weighted combination of the terms and their variants indexing the document. This technique appears to rank better most of the relevant documents and subsequently allows the selection of few documents to process by NLP techniques.

In this paper, we will present the QALC system that was developed for the QA track at TREC8 and TREC9. In the next section we describe the
architecture of the QALC system. Then we present the NLP based document selection process, and the experimentations that we made on the TREC corpus concerning the document selection. Finally, we discuss the results obtained by the QALC system and a number of related works.

## 2 OVERVIEW OF THE QALC SYSTEM

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 is a question about.

The analysis of a question relies on a shallow parser which spots discriminating patterns and assigns categories to a question. The categories correspond to the types of entities that are likely to constitute the answer to this 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 used in the process of document ranking and question-sentence pairing.

Documents are ordered according to the number and the quality of the terms 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 second 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. Such processes are time consuming, and we limit their application to a minimal number of documents. Thus, when the curve presents a high negative slope, we only select documents before the fall, otherwise a fixed threshold is used.

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. The three lists used for lexical lookup are 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 (1998) archive) and a list of 211,587 family names also from the CLR archive.

Finally, a pairing 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 and makes a combination of them. The QALC system proposes large 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 larger subpart without any terms. For a description of the whole system, see Ferret et al (2000).

## 3 DOCUMENTS SUBSET SELECTION

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 in the document and the original terms. The original terms are extracted from the questions. 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.

### 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 corpora. They are very close to those described by Justeson and Katz in (1995), but we do not include post-posed prepositional phrases. The pattern used for extracting terms is:

\[
\left((((\text{JJ}|\text{NN}|\text{NP}|\text{VBG})?((\text{JJ}|\text{NN}|\text{NP}|\text{VBG})(\text{NP}|\text{NN})))\right) \text{(VBD})(\text{NN})(\text{NP})(\text{CD})
\]

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 \text{name}_{\text{NN}} \text{o}_{\text{IN}} \text{the}_{\text{DT}} \text{US}_{\text{NP}} \text{helicopter}_{\text{NN}} \text{pilot}_{\text{NN}} \text{shot}_{\text{VBD}} \text{down}_{\text{RP}}\text{, 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 substrucures that correspond to an attachment of modifiers to the lefmost constituents (the closest one). For instance, the decomposition of \text{US helicopter pilot} into \text{helicopter pilot} and \text{pilot} is equivalent to extracting the subconstituents of the structure \text{[US [helicopter [pilot]]].}

### 3.2 NLP-based Indexing 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 those extracted from the questions, plus the other single content words of the questions. They 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 \textit{maker} is made of the nouns \textit{maker, make and remake}, and the verbs \textit{to make} and \textit{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 \textit{maker}, obtained from WordNet1.6, is composed of three nouns: \textit{maker, manufacturer, shaper} and the semantic family of \textit{car} is \textit{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 \textit{making many automobiles} as a variant of the term \textit{car maker}:

\[
\text{VM('maker')} \text{ RP? PREP? (ART (NN|NP)? PREP)? ART? (JJ | NN | NP | VBD | VBG)\text{[}\text{NS('car')}\text{]}
\]

\text{VM('maker')} is any verb in the morphological family of the noun \textit{maker} and \textit{NS('car')} is any noun in the semantic family of \textit{car}.

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

### 3.3 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:

\[
\text{LA092690-0038} \quad t(131,1) \\
\text{making many automobiles} \quad \text{NtoVSemArg}
\]

\(^1\text{NN are common nouns, NP proper nouns, JJ adjectives, VBG gerunds, VBD past particples and CD numeral determiners.}\)

\(^2\text{RP are particles, PREP prepositions, ART articles, and V verbs.}\)
means that the occurrence making many automobiles from document \( d = \text{LA092690-0038} \) is obtained as a variant of term \( i = 1 \) in question \( q = 131 \) (car maker) through the variation given in Section 5.1.

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 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. The following algorithm is used for calculating the cutoff threshold \( i_0 \) associated with the weighting scheme \( W \) of a given query \( q \):

\[
\text{If } \frac{W_i(d)}{W_{i-1}(d)} \leq 0.5 \text{ then } i_0 = 2 \\
\text{else} \\
i_0 = \min \left\{ i \in \left[ 3 \ldots 100 \right] \\
\frac{W_i(d) - W_{i-1}(d)}{W_i(d) - W_{i-2}(d)} \geq 2 \\
\wedge \frac{W_i(d)}{W_{i-1}(d)} \leq 0.8 \right\} \cup \{100\} \\
\]

Through this method, the threshold \( i_0 \) is 6 for question 854 (How do you abbreviate "Original Equipment Manufacturer"?), Figure 2.a) and 100 for question 842 (At what time of year is air travel at a peak?, Figure 2.b). As indicated by Figure 2.a, there is an important difference of weight between documents 6 and 7. The first documents have high weights due to the presence of the term Original Equipment Manufacturer, while documents after the sixth only contain few of the question words.

Finally, the system retains the \( i_0 \) best ranked documents with a minimum number set to 20.

4 EXPERIMENTS

One module of the QALC system is the selection, through a search engine, of documents that may contain an answer to a given question from the whole collection made of about 3 gigabytes.

4.1 Search Engine Evaluation

We tested three search engines with the 200 questions that were proposed at the TREC8 QA task. The first one is Zprise, a vectorial search engine developed by NIST. The second is Indexal (de Loupy et al, 1998), a pseudo-boolean search engine developed by Bertin Technologies. The third search engine is ATT whose results to the TREC questions are provided by NIST in the form

\[
3 \text{ We are grateful to Bertin Technologies for providing us with the outputs of Indexal on the TREC collection for the TREC9-QA question set.}
\]
of ranked lists of documents. We based our search engine tests on the list of relevant documents extracted from the list of correct answers provided by TREC organizers.

We compared the results given by the three search engines for a threshold of 200 documents. Table 1 gives the tests results.

Table 1. Compared performances of the Indexal, Zprise and ATT search engines

<table>
<thead>
<tr>
<th>Search Engine</th>
<th>Indexal</th>
<th>Zprise</th>
<th>ATT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions with relevant documents retrieved</td>
<td>182</td>
<td>193</td>
<td>194</td>
</tr>
<tr>
<td>Number of questions without relevant documents retrieved</td>
<td>18</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Total number of relevant documents that were retrieved</td>
<td>814</td>
<td>931</td>
<td>1021</td>
</tr>
</tbody>
</table>

The ATT search engine revealed itself the most efficient according to the following two criteria: the lowest number of questions for which no relevant document was retrieved, and the most relevant documents retrieved for all the 200 questions. We thus chose two different search engines for the TREC9 QA task. The first one is ATT, for the obvious reason that it provides the best results. The other is Indexal as Bertin Technologies is in the process of improving its system.

Since a search engine produces a ranked list of relevant documents, we had to define on the number of documents to retain for further processing. 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.

4.2 Document Selection Threshold
We carried out four different tests with the Zprise search engine, respectively with the top 50, 100, 200, and 500 retrieved documents. Table 2 shows the test results.

According to Table 2, the improvement of the search engine results tends to decrease beyond the threshold of 200 documents. For the TREC9 QA task, we then chose the threshold of the top 200 ranked documents that seemed to offer the best arrangement between the number of documents in which the answer may be found and the processing time by FASTR. This first subset of documents, ordered by the search engine, is then re-indexed and re-ranked in order to retain only a minimal number of documents.

Table 2. Number of questions with and without relevant documents retrieved for different thresholds

<table>
<thead>
<tr>
<th>Selection Threshold</th>
<th>Questions with relevant documents</th>
<th>Questions with no relevant documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>181</td>
<td>19</td>
</tr>
<tr>
<td>100</td>
<td>184</td>
<td>16</td>
</tr>
<tr>
<td>200</td>
<td>193</td>
<td>7</td>
</tr>
<tr>
<td>500</td>
<td>194</td>
<td>6</td>
</tr>
</tbody>
</table>

4.3 Document Ranking Evaluation
In order to evaluate the efficiency of the ranking process, we proceeded to several measures. First, we apply our system on the material given for the TREC8 evaluation, one time with the ranking process, and another time without this process. Thus the 200 documents were retained for each of the 200 questions. The system was scored by 0.463 in the first case, and by 0.452 in the second case. These results show that the results do not decreased when processing less documents and overall that the relevant documents are selected in the minimal subset. The interest to perform such a selection is also illustrated by the results given Table 3, computed on the TREC9 results. We see that the selection process discards documents for 50% of the questions (340 questions are processed from less than 100 documents). 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 is less than 100 documents is 37. These results are very interesting when applying such time-consuming processes as named-entities recognition and question/sentence pairing.

Table 3. Evaluation of the ranking process

<table>
<thead>
<tr>
<th>Number of documents selected by ranking</th>
<th>100</th>
<th>&lt;=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution among the questions</td>
<td>342 (50%)</td>
<td>340 (50%)</td>
</tr>
<tr>
<td>Number of correct answers</td>
<td>175 (51%)</td>
<td>200 (59%)</td>
</tr>
<tr>
<td>Number of correct answer at rank 1</td>
<td>88 (50%)</td>
<td>128 (64%)</td>
</tr>
</tbody>
</table>

The efficiency of our ranking process is also illustrated by Figure 3. It shows that the documents containing the right answer are better ranked by our process than by the ATT search engine. The graph shows the number of relevant
document, represented by their mean over the 682 questions of TREC9, among different documents subset sizes.

**Figure 3.** Mean number of relevant documents for different amounts of selected documents.

5 RESULTS AND DISCUSSION

Participants to TREC evaluation proposed up to 5 ordered answers to about 700 questions. The size of the answer should be either 250 or 50 bytes maximum. The score of a run is the mean reciprocal rank of the correct answers (0 otherwise) over the 682 questions.

We sent to TREC9 two runs which gave answers of 250 characters length. The first run used ATT as search engine, and for the second one, Indexal. Results are consistent with our previous analysis (see Section 4.1). Indeed, the run with ATT search engine gives slightly better results (0.407 strict) than those obtained with the Indexal search engine (0.375 strict). Table 4 sums up the number of answers found by our two runs.

Table 4. Number of correct answers retrieved, by rank, for the two runs at 250 characters

<table>
<thead>
<tr>
<th>Rank of the correct answer retrieved</th>
<th>Run using ATT</th>
<th>Run using Indexal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>216</td>
<td>187</td>
</tr>
<tr>
<td>2 to 5</td>
<td>159</td>
<td>185</td>
</tr>
<tr>
<td>Total of correct answers retrieved</td>
<td>375</td>
<td>372</td>
</tr>
<tr>
<td>No correct answer retrieved</td>
<td>307</td>
<td>310</td>
</tr>
</tbody>
</table>

In addition to the benefit for each participant from getting an evaluation of his/her own question-answering system, the TREC evaluation conference provides an interesting experimentation field for comparing the efficiency of the methods used by various QA systems. Even if the basic components of QA systems are about the same, methods used are rather different.

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. Some of them just get the ranked list of documents retrieved by ATT, while others used their own search engine or an available search engine that they can adapt.

For instance, Vicedo and Ferrandez (2000), whose system uses several NLP techniques, only keep the first fifty ranked documents from ATT. Their system solves pronominal anaphora in the documents, and builds the semantic context representation of the answer by means of WordNet. Their best score is 0.356 for answers of 250 characters length. Litkowski (2000) builds a semantic representation of the documents sentences by means of a syntactico-semantic analysis. They only process the 20 top documents from ATT. Their best score is 0.296 for answers of 250 characters length.

Some other systems, which have their own search engine, just keep one or more relevant text paragraphs from each document retrieved. Kwok et al (2000), for instance, have an IR engine, namely PIRCS, that retrieves the top 300 sub-documents of about 300-550 words. Their best score is 0.464 for answers of 250 characters length. Harabagiu et al (2000) uses a boolean search engine whose results are post-processed in order to retrieve only text paragraphs defined by the presence of the question words within a window of about ten lines. Their system, namely FALCON, uses a semantic approach to perform the matching between the question and the answer. After the paragraph selection stage, a unification is searched between the semantic representation of the question and that of selected paragraphs. Furthermore, FALCON is the only system that adds a logical justification for the correctness of the answer. Their system was ranked first with a score of 0.760 for answers of 250 characters length.

6 CONCLUSION

It is well known that QA systems have to use NLP techniques that add semantic and pragmatic knowledge to numerical IR methods, thus leading to more efficient systems. As such techniques are time consuming, the corpus where the answer is sought should not be too large. In this paper, we presented a NLP-based document selection
process as well as the experiments that show its efficiency. This selection is processed on the documents retrieved by a conventional search engine. We preferred this solution to the one consisting of adapting a search engine to paragraph selection because it is independent of the search engine that is used. As such, our document selection process can easily be used after any search engine and thus could be used for instance when searching an answer on the Web. Furthermore, this document selection is the more important as the improvements that we want to bring to our system will essentially pertain to a semantic and pragmatic approach: a better use of the specific relations between words in WordNet, the introduction of a semantic representation of the expected answer type, and finally a justification for the correctness of the answer.

7 REFERENCES

I. Consortium for Lexical Resources, UPenn.
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.