# Finding an answer based on the recognition of the question focus

O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, I. Robba, A. Vilnat
LIR Group
LIMSI – CNRS (France)

## 1. Introduction

In this report we describe how the QALC system (the Question-Answering program of the LIR group at LIMSI-CNRS, already involved in the QA-track evaluation at TREC9), was improved in order to better extract the very answer in selected sentences. The purpose of the main Question-Answering track in TREC10 was to find text sequences no longer than 50 characters or to produce a "no answer" response in case of a lack of answer in the TREC corpus.

As QALC first retrieves relevant sentences within the document corpus, our main question was: how to find the answer in a sentence? This question involves two kinds of answer: a) it is better to know what you look for and b) you have to know the location of what you look for. The first case is solved by applying a question analysis process. This process determines the type of the expected answer in term of named entity. This named entity is searched for in the sentences. However, all answers cannot be expressed in term of a named entity. Definition questions or explanation questions for example demand phrases (noun phrases or verb phrases) as answers. So, after having studied the structure of subpart of sentences that contained answers, we defined criteria to be able to locate the precise answer within a sentence. These criteria consist in defining triplets composed of a question category, the question focus and an associated list of templates allowing the location of the answer according to the focus place in the candidate sentence.
In the following sections, we will detail this novel aspect in our system by presenting the question analysis module, the different processes involved in the answer module and the results we obtained. Before, we give a brief overall presentation of QALC.

## 2. The overall architecture of QALC

The basic architecture of QALC is composed of different modules, one dedicated to the questions, one to the corpora, and a last module in charge of producing the answer. Each of these main modules is decomposed in several processes (see Figure 1).

The system is based on the following modules:

- Question module. This module regroups a question analysis process and a term extractor. The analysis of the questions relies on a shallow parser (Aït-Mokhtar 1997) in order to extract several pieces of information from the questions:
  - an answer type that corresponds to the types of entities which are likely to constitute the answer to this question.
  - a question focus: a noun phrase that is likely to be present in the answer
  - a question category that gives clues to locate the answer

The term extractor is based on syntactic patterns that describe compound nouns. The maximal extension of these compounds is produced along with the plausible sub-phrases. All the noun phrases belonging to this maximal extension are also produced.

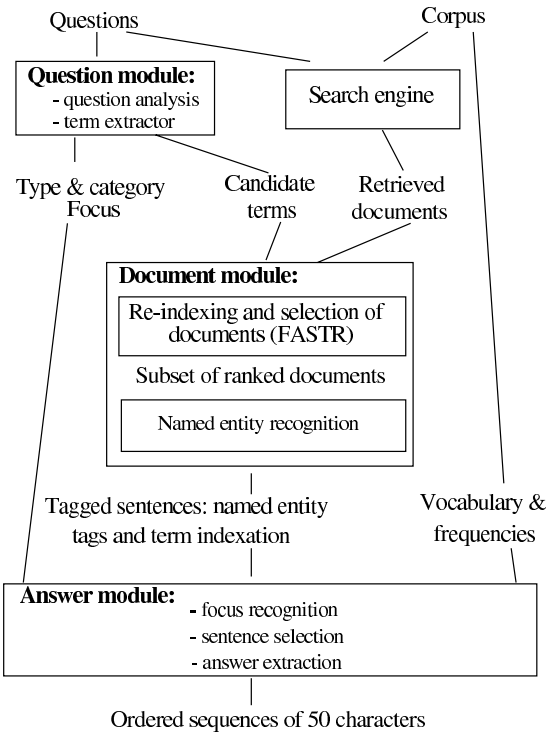


Figure 1: QALC architecture

- Document module. We use the outputs provided by NIST, resulting from the application of the ATT search engine. The 200 best documents are re-indexed by Fastr (Jacquemin 1999), a shallow transformational natural language analyzer that recognizes the occurrences and the variants of the terms produced by the term extraction process. Each occurrence or variant constitutes an index to the document that is ultimately used in the process of document ranking and in the process of question/document pairing. These indexes allow QALC to reorder the documents and entail the selection of a subpart of them (Ferret & al. 2001). A named entity recognition process is then applied on the resulting sets of documents.
- Answer module. This module relies on two main operations: the sentence selection and the answer extraction. All the data extracted from the questions and the documents by the preceding modules are used by a pairing module to evaluate the degree of similarity between a document sentence and a question. The answers are then extracted from the more relevant sentences according to several criteria:
  a) the presence of the expected answer type or not,
  b) the focus recognition in the sentence
  c) the category of the question and its associated patterns.

## 3. Natural Language Question Analysis

Question analysis is performed in order to assign the questions some features that will be used in the answer module. In view of a better search for the response, question analysis has to give as much information as possible. In our Trec9 system, this analysis allowed the prediction of an answer type, when it was a named entity (for instance, ORGANIZATION). In our Trec10 system, question analysis still allows the prediction of a named entity answer type but also the prediction of a more general answer type. Moreover, question analysis provides new information: the question focus and the question category.

### *3.1 Answer Type*

The question analysis module tries to assign to each question an answer type, which may be a named entity or a more general type. In the first case, the module tries to find if the answer type corresponds to one or several named entity tags sorted by importance order. The named entity tags are hierarchically organized within 17 semantic classes (Ferret and al. 2000). For example:

> Question: Who developed the Macintosh Computer?
> Named Entity List = PERSON ORGANIZATION

In addition, question analysis tries to deduce a more general type. It means to find a noun or a noun phrase that corresponds to an entry in the WordNet lexical base. For example,

> Question: What metal has the highest melting point?
> General Type = metal
>
> Question: What is the name of the chocolate company in San Francisco?
> Named Entity List = ORGANIZATION
> General Type = company

### *3.2 Focus*

Next, question analysis tries to deduce the question focus, which corresponds to a noun or a noun phrase that is likely to be present in the answer. For each question, we will determine a focus, a focus head (the main noun) and the "modifiers" of the focus head (adjective, complement...). For example:

> Question: Who was the first governor of Alaska?
> FOCUS = the first governor of Alaska
> FOCUS-HEAD = governor
> MODIFIERS-FOCUS-HEAD = ADJ first, COMP Alaska

### *3.3 Question Category*

The detection of question category gives us a clue to find the location of the answer in a candidate sentence. Each question category corresponds to a syntactic pattern. The question category is the "syntactic form" of question. For example:

Question: What does a defibrillator do?
Category = WhatDoNP

Question: When was Rosa Park born?
Category = WhenBePNborn

After studying the questions of TREC8 and TREC9 along with the sentences containing an answer, we found more than 80 question categories. This repartition of questions in categories enables the definition of rules to find the focus and answer type information.

### *3.4 Criteria for question analysis*

To find all these different items of information, we used syntactic and semantic criteria. Syntactic information is provided by a shallow parser (Aït-Mokhtar 1997) applied to all questions. Thus, QALC obtains a segmentation of each question into chunks and a set of syntactic relations between them. But often, the shallow parser is not appropriate for analyzing question, so we had to recapture parse mistakes.
Rules to find the focus, the category and the answer type were written from the syntactic representation of the question. Semantic criteria are extracted from the WordNet lexical base to improve the named entities glossary, and to find a more general answer type.
For the TREC10 questions, our question module finds 85 % of the correct focus, 87 % of correct general answer type and 90.5 % of correct named entity type.

## 4. Focus recognition

The focus of a question is structured as follows: (a) the head of the focus, (b) a list of modifiers. QALC tries to locate this focus in the sentences of the selected documents. It first detects the head of the focus, and then identifies the noun phrase in which the head is enclosed. To determine the frontiers of this noun phrase, we define a local grammar for the NP in English. This grammar relies on the tagging made by the Tree-Tagger (Smidt&Stein 99). For example, for the question 827:

*"Who is the creator of the Muppets?",*

the focus is "the creator of the Muppets", with the head : "creator".
In a document, we found the following NP:

*late Muppets creator Jim Henson,*

which fits the expression:

Adjective + Plural Noun + Noun + Proper Noun + Proper Noun

We also look for NPs containing synonyms of the question focus head. These synonyms are determined by FASTR. When the recognition of a focus in the question failed, QALC looks for the proper nouns in the question, and it tries to recognize NPs containing these proper nouns.
When these NPs are delimited, we associate them a score. This score takes into account the origin of the NP and the modifiers found in the question: when the NP contains the modifiers present in the question, its score is increased. The best score is obtained when all of them are present.
In the example on question 827, the score is maximal: the NP has been obtained directly from the focus of the question, all the significant words of the focus are present: "*creator*" and "*Muppets*".

When the NP is obtained with a synonym of the focus head, the score is only slightly decreased, and a little more when it is obtained via a proper noun. However the scoring algorithm always takes into account the ratio between the number of words present in the question phrase and in the document noun phrase.

For example the score assigned to the NP:

"*their copy of the 13th century Magna Carta*" obtained for the question 801 :

"*Which king signed the Magna Carta*", has a lower score because it has not been obtained from the focus ("*king*"), but from the proper noun "*Carta*", even if it contains all the words of this proper noun phrase: "*Magna*" and "*Carta*".

For each sentence of the selected document, QALC tags all the relevant NPs following the preceding algorithm, with the associated scores. It only keeps the NPs obtaining the best scores, which in turn provides an evaluation of the relevance of the sentence, which will be used in the pairing module in charge of the sentence selection.

## 5. Sentence selection

In our system for TREC 10, the pairing module achieving the selection of a set of sentences that possibly contain the answer to a question is based on the same principle as the pairing module used in our TREC 8 and TREC 9 systems: it compares each sentence from the selected documents for a question to this question and constantly keeps in a buffer the $N^1$ sentences that are the most similar to the question. This comparison relies on a set of features that have been extracted both from the questions and the sentences of the selected documents:

- terms;
- focus;
- named entities;
- scattering of terms in the sentence.

A specific similarity score is computed for each of these features. The last feature enables the module to decide between two sentences having the same score for the first three features.

We tried different weighting schemes for terms (Ferret & al 2000). The one we choose here was to sum the weights of the terms of the question that are in the document sentence. A term weight integrates its normalized information with regards to a part of the *QA* corpus (vocabulary frequencies in figure 1) and the fact that it is or not a proper noun.

The term score is combined with the focus score and the resulting score constitutes the first criterion for comparing two document sentences *S1* and *S2*: if *S1* has a combined score much higher than *S2*[2], *S1* is ranked on top of *S2*. Otherwise, the named entity score is used in the same way. It evaluates to what extent a named entity in a document sentence can fit the target of a question when the expected answer is a named entity. This measure takes into account the distance of their two types in our named entity hierarchy.

When the two preceding criteria are not decisive, the first criterion is used once again but with a smaller threshold for the difference of scores between two sentences. Finally, if there is still an uncertainty, the module ranks first the sentence that has the shortest matching interval with the question. This interval corresponds to the shortest part of the sentence that gathers all the terms of the question that were recognized in it.

---

[1] *N* is at least equal to 5. The selected sentences are ranked according to their similarity to the question.

[2] « Much higher » means that the difference of scores for *S1* and *S2* is higher than a fixed threshold.

## 6. Answer extraction

The extraction process depends on whether the expected answer type is, or is not, a named entity. Indeed, when the answer type is a named entity, the extraction consists of the location of the named entity within the sentence. Thus, it mainly relies on the results of the named entity recognition module. On the other hand, when the answer type is not a named entity, the extraction process mainly relies on the recognition of the question focus, as it consists of the recognition of focus-based syntactic answer patterns within the sentence.

### *6.1. Named entity extraction*

When the question allows the system to predict the kind of expected answer in term of a named entity type, the extraction of the answer is based on this information. This process looks for all the expressions tagged with the searched type. If several such expressions exist, we choose the closest to the focus, if it was recognized in the sentence, otherwise the first one. When there is no named entity of the type desired, QALC generalized the searched type using our own hierarchy. By this way, when looking for a person, QALC will look for a proper name, or look for a number instead of a length, etc.

### *6.2. Answers of type "common noun or verb phrase"*

When the expected answer type is not a named entity, the QALC system locates the very answer within the candidate sentence through syntactic patterns. Syntactic patterns of answer include the focus noun phrase and the answer noun phrase, which can be connected by other elements such as comma, quotation marks, a preposition or even a verb. Thus, a syntactic pattern of an answer always includes the focus of the question. As a result, the focus has to be determined by the question analysis module in order to enable the QALC system to find a common noun or verb phrase as answer.

If we consider the following question (n°671):

*" What do Knight Ridder publish? "*

The focus of the question, determined by the rules of the question analysis module, is "*Knight Ridder*". This question pertains to the question type What-do-NP-VB, with "*Knight Ridder*" as NP and the verb "*publish*" as VB.

One answer pattern applying to this category is called FocusBeforeAnswerVB and consists of the following syntactic sequence:

NPfocus Connecting-elements NPanswer

The NPfocus is the noun phrase corresponding to the question focus within the sentence-answer. It is followed by the connecting elements, then by a noun phrase that is supposed to contain the very answer. The connecting elements mainly consist of the question verb (VB in the question type).

The following answer, which was found in the documents corpus, fits with the FocusBeforeAnswerVB pattern:

" *Knight Ridder publishes 30 daily newspapers ...* ",

This answer was extracted from the following sentence:

" *Knight Ridder publishes 30 daily newspapers, including the Miami Herald and the Philadelphia Inquirer, owns and operates eight television stations and is a join venture partner in cable television and newsprint manufacturing operations.* ".

We saw, in section 3.3, that about 80 question categories were determined from the corpus. Among them, about 45 do not expect a named entity as answer, and thus need syntactic patterns. For each of those question types, we built syntactic patterns. The different patterns, as well as the different question types, were empirically determined from corpus analysis. The corpus consisted of the questions and answers provided after the TREC8 and TREC9 conferences. We considered 24 patterns. The number of patterns for each question type varies from 2 to 20, with an average of 10 patterns for each question category. Thus, several question types share the same pattern.
The difficulty in finding syntactic patterns varies according to the question type. This difficulty is partly due to the small number of some question types within the corpus, and, for the most part, to the grammatical diversity of the answers. For example, there is few " Why " questions (4) and few " How verb " questions (4), such as " Why can't ostriches fly? " (n° 315) and " How did Socrates die? " (n° 198). Moreover, answers to those questions can hardly be reduced to a pattern. We also hardly found grammatical regularities in the answers to the " What-GN-be-GN " questions, such as " What format was VHS's main competition? " (n° 426) or " What nationality was Jackson Pollock ? " (n° 402) for instance. Indeed, depending on the situation, it is the first NP (" format " or " nationality ") or the second NP (" VHS " or " Jackson Pollock "), which plays the main role in the pattern.

## 7. Results and Analysis

The three runs that we sent to TREC10 come from the same selection of the top ten more relevant sentences. Those runs are the result of three different weighting schemes for the top ten answers, weighting that thus ranked them differently.

| | run QALIR1 | run QALIR2 | run QALIR3 |
|---|---|---|---|
| strict evaluation | 0.181 | 0.176 | 0.167 |
| lenient evaluation | 0.192 | 0.188 | 0.179 |

### *7.1 Top five answer selection*

For each question, the pairing module presented section 5 selects ten sentences. Hence, the final problem is to choose five ranked answers among them. Three strategies were implemented:

- selecting the first five answers according to the order given by the pairing module. This is more precisely the order of the selected sentences from which the answers were extracted;
- selecting the first five answers according to the order given by the answer extraction module. This module ranks its answers according to the patterns that were applied for extracting them. The answer score is the highest when the pattern applied is the most typical for the question category;
- a mixed strategy that merges the two previous lists : following the order of the two preceding lists, one answer is alternately taken from one list and the following from the other list until having five answers.

No specific processing was done for detecting that an answer cannot be found in the *QA* corpus: a « no answer » answer is provided when the pairing module cannot select at least one sentence or when the answer extraction module cannot apply a syntactic pattern in the selected sentences.
For providing a « final answer », we only worked on detecting when the answers to a question are globally not sure. Otherwise, we considered that the first answer of our list (rank 1) was the final answer. Comparing the lists given by the two first strategies of answer selection did the detection of the unsure cases: if the two lists were too different according to a similarity measure, the question was marked as unsure. This measure takes into account the differences concerning both the presence of an answer and the rank in the lists.

### *7.2 QALC performances according to the expected answer type of the question*

We previously distinguished questions that expect a named entity as answer, from questions that expect a noun or verb phrase. Indeed, when the answer is a named entity, the location of the answer within the sentence is facilitated by the presence of the named entities tags within the documents. In fact, QALC obtains better results regarding the named entity questions than other questions. Actually, while 46.5% of the TREC10 questions expect a named entity as answer, 56% of the correct answers respond to a named entity question. The named entity questions are questions that have been recognized as such by the question analysis module. On the other hand, QALC achieves not so good performances regarding the named entity questions in TREC10 (31.4% of correct answers) than in TREC9 (39.3% of correct answers).
Anyway, the QALC system performs better than its previous version concerning the questions that expect a noun or verb phrase. Indeed, 21.3% of those questions have correct answers in TREC10 evaluation, for only 10% in TREC9.

## 8. Conclusion

In this article, we focused on the extraction of the precise answer. As we described above, the QALC system first selects the sentences that respond to the question, and then, extracts the precise answer from them. This principle is efficient when the selected sentences have weights very different from each other. In this case, the answer has a high probability to be in one of the top ten sentences. But, when many sentences have close weights, the answer may be just as well in the fiftieth sentence as in the first one. To face up this situation, another strategy has to be carried out.
Another problem we met with is the setting-up of the syntactic patterns of answer. Those patterns are drawn from question-answer corpora, and thus require large corpora to be efficient. This is not the only difficulty: answers to some categories of question can hardly be reduced to patterns. We have to find another solutions concerning those categories. One solution we tested uses WordNet. Indeed, we noticed that knowing the expected answer type (when it does exist) facilitates the recognition of the answer within the sentence. At present, QALC recognizes named entities such as persons, cities, states, organizations, and numbers such as financial amounts and physical magnitudes. However, the QALC question analyzer is able to recognize more answer types than those that are now tagged by the named entity recognition module. For instance, the question n° 380 "*What language is mostly spoken in Brazil* " expects a language name as answer. As this type is not tagged within the documents, QALC is not able to locate it directly within the sentences. Thus, we tested the use of WordNet so as to validate the answer. " *Portuguese* ", which is the answer in our example, is a member of the hyponym hierarchy for « language » in WordNet. This gives us a way to validate the answer: if one of the answers found

by QALC has as hypernym the answer type recognized by QALC within the question, thus QALC would select this answer.

## References

(Aït-Mokhtar 1997) Aït-Mokhtar S. and Chanod J. (1997), Incremental Finite-State Parsing, In Proceedings of *ANLP-97*, Washington.

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

(Ferret & al 2001) O. Ferret , B. Grau , M. Hurault-Plantet, G. Illouz, C. Jacquemin, Document selection refinement based on linguistic features for QALC, a question answering system, RANLP 2001, bulgarie

(Jacquemin 1999) Jacquemin, C., Syntagmatic and paradigmatic representations of term variation. In Proceedings, ACL'99, University of Maryland, 341-348.

(Schmid 1999) H. Schmid, Improvements in Part-of-Speech Tagging with an Application To German. In Armstrong S., Church K.W., Isabelle P., Manzi S., Tzoukermann E., and Yarowsky D. (eds) Natural Language Processing Using Very Large Corpora, Kluwer Academic Publishers, Dordrecht.