The question answering system QALC at LIMSI: experiments in using Web and WordNet

G. de Chalendar, T. Dalmas, F. Elkateb-Gara, O. Ferret, B. Grau, M. Hurault-Plantet,
G. Illouz, L. Monceaux, I. Robba, A. Vilnat
LIMSI – CNRS (France)

1. Introduction

The QALC question answering system at LIMSI (Ferret et al, 2001) has been largely modified for the TREC11 evaluation campaign. Architecture now includes the processing of answers retrieved from Web searching, and a number of already existing modules has been re-handled. Indeed, introducing the Web as additional resource with regard to the TREC corpus, brought us to experiment comparison strategies between answers extracted from different corpora. These strategies now make up the final answer selection module.

The answer extraction module now takes advantage of using the WordNet semantic data base, whenever the expected answer type is not a named entity. As a result, we draw up a new analysis for these question categories, just as a new formulation of associated answer extraction patterns. We also changed the weighting system of the sentences which are candidate for answer, in order to increase answer reliability. Furthermore, the number of selected sentences is no longer decided before extraction module but inside it according whether the expected answer type is a named entity or not. In the last case, the number of selected sentences is greater than in case of a named entity answer type, so as to take better advantage of the selection made by means of extraction patterns and WordNet.

Other modules have been modified: the QALC system now uses a search engine and document selection has been improved through document cutting into paragraphs and selection robustness improvement. Furthermore, named entity recognition module has been significantly modified in order to recognize precisely more of them and decrease ambiguity cases.

In this paper, we first present the architecture of the system. Then, we will describe the modified modules, i.e. question analysis, document selection, named entity recognition, sentence weighting and answer extraction. Afterwards, the strategies of final answer selection of the new module will be described. Finally, we will present our results, ending with some concluding remarks.

2. Architecture

QALC system core is made of classical following modules: question analysis, document selection, named entity recognition and answer extraction. As input of the system, we use the same TREC11 question set, but two different corpora, on the one hand the TREC11 corpus and on the other hand the Web. Figure 1 shows the architecture of the system. In this figure, various arrows indicates TREC11 corpus processing and Web documents processing. Selected answers provided by the two processing chains are merged in the final answer selection module (see section 4).

The Web chain of QALC is nearly the same as the classical one, except that the answers are looked up in documents gathered from the Web instead of in the QA Track corpus. The idea behind that is, as in (Soubbotin and Soubbotin, 2001), that there is a great chance to find the answer to a question in the Web in a shape similar to the one of the question itself, but in an affirmative form. This hypothesis is based on the huge quantity of documents directly available on the Web and thus on their high redundancy. So, for the question “When was Wendy’s founded?”, we expect to be able to find a document containing the answer in the form: “Wendy’s was founded on …”. We want to search the Web for strings with exact match. In the previous example, we don’t want to search for documents containing one of the words of the

query. Rather than that, we want to search for the exact phrases. For such a goal, a boolean search engine able to accept queries made of multiple words terms is necessary. Google is such a tool.

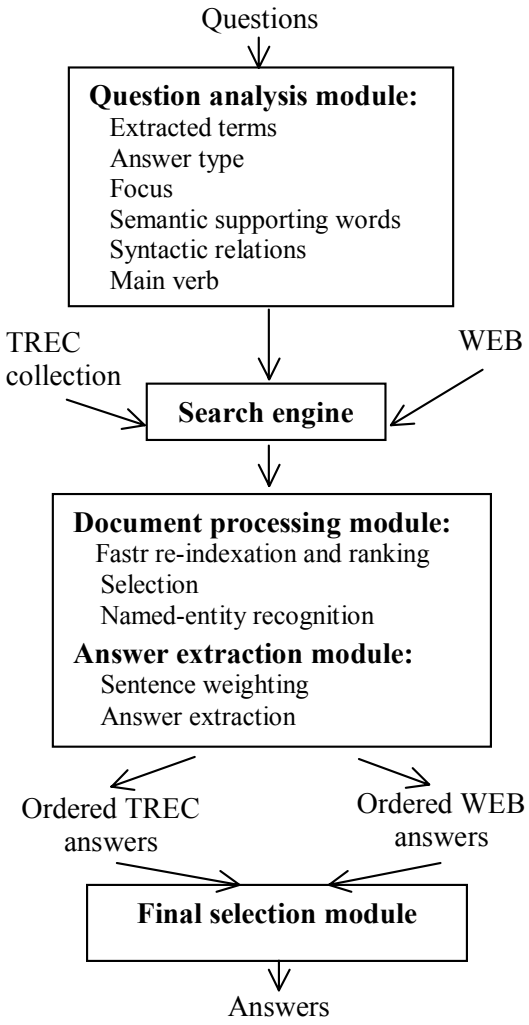

Figure 1. QALC System Architecture

The most important part for this chain is the rewriting of the question. It uses the results of the questions analysis and rewriting rules manually written from the study of the TREC 9 and 10 questions. The questions are categorized in function of their type and their category. If these two indices are not sufficient to adequately rewrite a question, a lexical marker is used. For example, the question "What continent is Egypt on?" has for type LOCATION-STATE and is of the category WhatGNbeGN. The focus of this question is "Egypt" and the kind of location (typeGen) to find is a "continent". The rule used to rewrite this question is:

(type = LOCATION-STATE) & (category = WhatGNbeGN) & (lexical-modifier = on)

=> "<focus> <verb> on" <typeGen>

So, the question will be rewritten in: ""Egypt is on" continent".

Here, the quotes are important, they indicate to Google to search for documents containing this exact string. This query finds 50 answers on Google and the first one contains the string: “Even though Egypt is on the continent of Africa…”.

We try to make the rules as specific as possible in order that the documents gathered by the queries they generate have the highest probability to contain an answer to the question. The drawback of this approach is that the probability not to find any document corresponding to the query increases with the specificity of the rules. To counter this tendency, we systematically propose several rules for each question type ordered by descending specificity. The most general one is simply the question without any edition.

The remaining of the Web chain is exactly the same as the classical one. At the end of the chain, we have one or more propositions of solutions. We cannot submit them directly as answers in the TREC QA track as they are not supported by documents of the TREC11 corpus. So, they are used to do a second path in the chain, this time querying the QA corpus with our search engine with queries enriched by the Web candidate answers. The answers that are found in the Web and that are confirmed by the corpus see their score greatly increased compared to the answers found only in the corpus.

In the future, we plan to use the work done with the Web chain more completely by using the rewritten questions to search the TREC corpus. This would improve the results of the search engine for the documents containing answers expressed similarly to the question. Such question-document pairs are certainly less numerous than in the Web but there are certainly a certain amount of them and we should not miss them.

3. QALC main processing chain

3.1 Question analysis

Question analysis is performed with the aim of extracting from the question the essential information required for retrieving an answer within a document sentence. When the expected answer type is a named entity, then essential information is the named entity type itself. In this case, analysis is based on the recognition of lexical and syntactical clues within the question that determine the named entity type of the answer. This year, we improved our system by analysing more precisely some named entity types, in particular physical numerical entities, and by adding some units to those already used.

But, if the expected answer is not a named entity, then the essential information are answer extraction pattern, and semantic relations between the answer and words of the question. Thus, the goals of the analysis are first to determine the question type in order to associate with it an extraction pattern, and then to build a validation schema that will instantiates the parameters of the pattern. The validation schema is built from the semantic supporting words of the question, i.e. the words that will have semantic relations with the answer within the sentence. For this purpose, a local syntactical analysis first locates within the question the grammatical features of a same question type. This analysis allows the system to recognize the question type and the semantic supporting words that will be used to produce the validation schema. Then, the validation schema is built from a set of production rules that uses the analysis results.
Let us consider for example the question 1505:

Question : What is the currency used in China?
The analysis of the question gave the following features structure:
Question qMoney :
Answer extraction pattern: aMoney
Semantic supporting words:
Answer type: currency
Focus: China

From this analysis results, the production rules then gave the following validation schema:

Hypernym: monetary unit
Country: China

This schema will be subsequently used by the extraction pattern named *aMoney* (see section3.5.2).

3.2 Document selection

QALC system uses two subsequent document selections, the first one performed by a search engine, and the second one performed by FASTR, adding variants of the question words as new criterion. For the first selection, we used MG[1], a boolean search engine. Our choice was to cut out the collection documents in paragraphs of approximately the same size. By this way, we add to the selection criteria a criterion about their proximity. The query was built from the question analysis results, by combining lemmatized and non lemmatized forms of the content words and adding the focus terms in order to reinforce this selection criterion.

Concerning the second selection, we made some improvement. The way the documents that are likely to contain an answer to a question are selected from the results of the search engine is globally the same in this version of QALC as in its previous versions. First, a set of one-word and multi-word terms is extracted from the question by a pattern-based termer. Then, the documents retrieved by the search engine are indexed by the FASTR tool (Jacquemin, 2001), which recognizes in the documents the terms extracted from the question, as well as their morphological, syntactic or semantic variants. Each recognized term is weighted according to the kind of variant it is and a score is computed for each document by aggregating the weights of its indexes. Finally, a restricted set of documents is selected when a significant shift in the documents' scores is found. Otherwise, a fixed number of them, in this case 100, is taken.

For the TREC11 evaluation, we specifically took into account a problem we had noticed in the previous evaluations. FASTR was designed for recognizing complex variants of terms in the field of terminology but not as a robust tool for information retrieval. For instance, the errors of the morpho-syntactic tagger we use, in this case the TreeTagger tool (Schmid, 1999), have a significant impact on its results. As a consequence, FASTR misses some non-variant occurrences of questions' terms that can be recognized by a basic term matcher. More precisely, we found that for one-word and multi-word terms without variation the recall of FASTR is equal to 71%. This evaluation was done with the documents selected for the 500 questions of the TREC11 evaluation. The reference was set by a term matcher working from the results of TreeTagger.

	QALC TREC 10	**NIST-100**	**QALC TREC 11**
selected documents with an answer (nb)	2041	2479	2313
selected documents (nb)	30992	49900	34568
recall (%)	46.0	55.8	52.1
precision (%)	6.6	5.0	6.7
selected documents with an answer - variation (%)	reference	+ 21.5	+ 13.3
selected documents - variation (%)	reference	+ 61.0	+ 11.5

Table 1: Document selection results for TREC10 questions

In order to improve the selection of documents, the version of QALC for the TREC11 evaluation combined the results of FASTR and those of our basic term matcher: the terms found by the term matcher are added to those found by FASTR and the doubles are discarded. The impact of this change is shown in Table 1. NIST-100 corresponds to a selection module that would always take the first 100 documents returned by the search engine. The results of Table 1 are based on the list of judgments about participants'

[1] Managing Gigabytes is an open source indexing and retrieval system. Its homepage may be found at: http://www.cs.mu.oz.au/mg

answers given by NIST for the TREC10 evaluation. The precision measure is the ratio between the number of selected documents that actually contain an answer and the number of documents selected by the considered system. The recall measure is the ratio between the number of selected documents that actually contain an answer and the number of documents found by at least one participant of TREC10 and that contain an answer. Table 1 shows that the combining of two term recognizers, a basic one and a more sophisticated one, is an interesting strategy as it makes both recall and precision increase. Moreover, a significant number of new relevant documents are found while the number of documents to process by the answer extraction modules only increases linearly.

3.3 Named entity recognition

The named entity module identifies named locations, named persons, named organization, dates, times, monetary amounts, measures and percentages in text. For Trec11, we have developed a new system for locations, persons and organizations recognition. In order to identify these entities, our system uses hand-made rules in which we specify named entities structure in term of text tokens and what we can find about them from resources such as tagger, morphosyntactic analyzer and knowledge base of names, clue words and abbreviations.

The system performs in three stages:
- Preprocessing: we use treetagger to tokenize the input text and tag them with a syntactic and typographic category.
- Database lookup: for each token or group of token, we check in a list of known names, abbreviations and clues. If a token is found in a database, this information is added to token feature.
- Named Entity analyzer uses language specific context-sensitive rules based on word features recognition pattern matching. For each token, we look for the longest pattern of token features that matches with pattern rules.

Features used for tokens are:
- Lookup in knwoledge base of names (persons, organizations, locations).
- Lookup in knowledge base for names, organizations and locations clues.
- Lookup in knowledge base for firstnames abbreviations.
- Typographic features (Capitalization, Roman characters, etc).
- Syntactic category.
- Lemma.

For TREC11 we wrote:
- 7 rules for Organizations,
- 9 rules for Locations,
- 7 rules for Persons.

Here is a example of detecting location entity:

If the (syntactic category of token (or group of tokens) is "Proper noun") and these tokens are followed by a token found in State clues Database, we identify all these tokens as a location.

3.4 Document sentence weighting

All the sentences in selected documents are analysed in order to give them a weight that reflects both the possibility that the sentence contains the answer, and the possibility that the QALC system locates the answer within the sentence. The criteria that we used produce simple processing, and are closely linked with the basic information extracted from the question. The resulting sentence ranking do not have to miss obvious answers. Thus, the main goal of this analysis is to assign a higher weight to sentences which contain most of obvious information. In return, all sentences are kept for subsequent processing, except those which do not contain any word from the question. Our aim is that the answer extraction modules can raise to an upper rank a lower weighted answer thanks to added specific criteria.

The criteria that we retained use the following features retrieved within the candidate sentence:

1. question lemmas, weighted by their specificity degree,
2. variants of question words,
3. exact words of the question,
4. mutual closeness of question words,
5. word whose type is the expected answer named entity type.

The specificity degree of a lemma depends on the inverse of its relative frequency within a large corpus. The criterion of mutual closeness of question words is the closeness between them arranged in pairs in the sentence.

First we compute a basic weight of the sentence based on the presence of question words within the sentence, and then we add weights from the other criteria. The computation of the basic weight of a sentence is made from lemmas (or from words if the word is unknown for the tagger), and their specificity degree. Some words are not taken into account, i.e. determinants or prepositions, transparent nouns, and auxiliary verbs. According to Fillmore (2002), a transparent noun is a noun whose complement is semantically more relevant that the noun itself. For instance, the word *name* is transparent in the question 1396 *What is the name of the volcano that destroyed the ancient city of Pompeii?*, and *volcano* is the semantically relevant noun. We made an a priori list of such words.

Thus, the basic weight of a sentence is given by:

$$P = (dr_1 + \ldots + dr_i + \ldots + dr_m) / (dq_1 + \ldots + dr_j + \ldots + dq_n)$$

with:
dr_i: specificity degree of a lemma from the question found in the sentence,
dq_j: specificity degree of a lemma of the question,
m: number of lemmas found in the sentence,
n: number of lemmas in the question.

Each lemma is taken into account only once even if it occurs more than once in the same sentence. If a word from the question is not found in the sentence, but a variant of it, half of the specificity degree of the word is added to the basic weight of the sentence. As the basic weight is relative, its maximum is equal to one. We bring it to 1000 for convenience. We subsequently add an additional weight to this basic weight for each additional criterion that is satisfied. Each additional weight cannot be higher than about 10% of the basic weight.

3.5 Answer extraction

3.5.1 Named entity answer type

If the expected answer type is a named entity, then selected answers are these words within the sentence that correspond to the expected type. In order to extract the answer, the system first selects, among the sentences provided by the sentence weighting module, the ten sentences that have the best weight. Then, it computes additional weights taking into account:

1. exact or generic named entity type of the answer,
2. location of the potential answer with regard to the question words within the sentence,
3. redundancy of an answer.

A sentence may not contain a word corresponding to the expected named entity type, but to a more generic one, for instance NUMBER instead of DATE. In that case, a lower weight will be given to the generic type. An additional weight is then given to potential answers closest to the question words. This closeness is computed with regard to the barycentre of the question words within the sentence. If there is more than one potential answer in the sentence, the one, that will be selected, is the closest to the question words. Finally, if a same potential answer is retrieved more often than others in the ten sentences, then it is assigned with an additional weight. These criteria allow the system to rank better a potential answer that had not the best weight as sentence. For instance:

Question 1475: *Who was the first person to reach the south pole?*
Candidate sentences:

1210 NYT19981103.0190 < PERSON> Diana Preston <e_enamex> s absorbing and moving story of the attempt by the British explorer < PERSON> Robert Falcon Scott <e_enamex> to be the first to reach the South Pole shows that that reverence for the noble failure is not unique to Japan.

1185 NYT19991028.0488 The truly adventurous go all the way to the South Pole , first reached in 1911 by< PERSON> Roald Amundsen <e_enamex> , a Norwegian .

1165 NYT19991129.0264 The Norwegian < PERSON3> Roald Amundsen <e_enamex> led the first successful expedition to the South Pole , reaching it on Dec. 14 , 1911.

The weight of each sentence is indicated in the first column. The lower weight of the second sentence comes from the exact word criterion (*reached* instead of *reach*). On the other hand, the third sentence has a lower weight due to less closeness of words. After having computed the added weights, we then obtained the following answers:

1365 NYT19991028.0488 Roald Amundsen
1267 NYT19981103.0190 Robert Falcon Scott

In that case, the redundancy criteria brought the correct answer to the first rank.

3.5.2 Common noun or phrase answer type

Each candidate sentence provided by the sentence selection module is analysed using the extraction pattern determined by question analysis. This pattern uses the associated validation schema to instantiate its parameters.

Extraction patterns are composed of a set of constraint rules on the candidate sentence. Rules are made up of syntactic patterns that are used to locate potential answer within the sentence, and of semantic relations that are used to validate answer. Syntactic patterns locate connecting words and simulate possible paraphrases of the answer. Semantic constraints are proved using WordNet. Extraction patterns are implemented through automata whose transitions are a set of functional constraints that have to be satisfied by the feature structure, representing the sentence, being valued. Potential answers are weighted according to the satisfied constraints. The weight amount depends on the reliability of the constraint. For instance, the hypernym relation constraint is given a high weight as it is a reliable relation.
Let us consider the example analysed in section 3.1:

Question: What is the currency used in China?
Answer: yuan
Extracted from the sentence:
The central bank governor acknowledged that the Renminbi yuan , China s currency , is now facing pressure for further appreciation due in part to growing foreign exchange reserves which reached 126 billion US dollars at the end of July .

The retrieved answer satisfies the following constraints:

Syntactic extraction pattern:
Answer , Country <currency | money>
Semantic validations:
Monetary unit *is hypernym of* yuan
Answer *has in gloss* Country

With *Country* instantiated by *China*, according to the validation schema.

At the output of the sentence weighting module, 12 candidate sentences had the same highest weight equal to 1120. Among these sentences, 5 contained the correct answer. The answer extraction criteria, that the QALC system used, thus allowed it to select the correct answer.

4. Final answer extraction module

In TREC11 evaluation, we had to supply one answer per question and the set of 500 answers had to be ordered according to a confidence score. As explained before, this year, we elaborated two different search strategies: the main strategy searches the answers only in the TREC collection, while the Web strategy searches the answers on the Web and then tries to confirm these answers by locating them also in the TREC collection.

Moreover, these two strategies supply for each question, a set of answers (but we examine at most the first five), which are ordered according to the score they received during the sentence weighting and answer extraction processes. The role of the final selection is hence to choose a unique answer between these two sets. For this selection we worked out two different algorithms applying a sequence of rules that we briefly present here:

If the same first answer is found by the Web strategy and by the main strategy, this answer is returned with an augmented score, the score increase being more significant if the answer is not NIL.
If a first answer is found by the main strategy and if the first answer of the Web strategy is NIL, the main strategy first answer is returned with its original score (or conversely but only if the Web answer is confirmed).
If a first answer is found by the Web strategy, but if this answer is not confirmed and if the answer of the main strategy is NIL, the answer NIL is returned with its original score.

When any of the preceding rules can be applied, we tested two different algorithms; both of them take into account not only the first answers of the sets but also possibly the other answers of the two sets.

The first algorithm consists in increasing the score of the first answer of the main strategy, provided that this first answer is also present in the Web strategy results. If the first answer is not found in the Web results, the other way round, the first answer of the Web strategy is searched in the main strategy results. If the two searches fail the main strategy first answer is returned with its original score.

The second algorithm attributes a score to each couple (i,j), i being the position of an answer in the answer set of the main strategy, j being the position of an answer in the answer set of the Web strategy. The score is especially high since the two answers are equal. The answer of the couple obtaining the best score is finally returned with a score that is augmented according both positions: i and j.

Example:

Question 1806: When was the first heart transplant?
EN - Answers:
Answer 1: in 1979,
Answer 2: in 1967 …
Web - Answers:
Answer 1: in April 1985,
Answer 2: on December 3, 1967,
Answer 3: in 1968,
Answer 4: in 1967, …
Final – Answer: in 1967 obtained by couple (2,4)

5. Results

Table 2 presents the results we obtained in TREC11 evaluation for the three runs we submitted. As explained before, we try a new solution to select the pertinent documents, using another engine MG (run 3), with documents cut in paragraphs. We also combine FASTR with a basic term matcher to select documents (as illustrated in section 3.2). To evaluate the advantages of these strategies, we also use the results provided by NIST. So the two first runs are obtained with NIST search engine and the third one with MG. The first run considers FASTR alone, the second and third ones consider the combination of FASTR with a basic termer. The first run uses the first algorithm to choose the final answer (section 4), the two others use the second one. All of them take advantage of the results of the Web search.

	W	U	X	R	TREC11 score	NIST pattern score
Run 1 — NIST 1 & Web	342	21	7	130	0.485	0.567
Run 2 — NIST 2 & Web	336	20	11	133	0.497	0.587
Run 3 — MG & Web	330	20	11	139	0.488	0.572

Table 2: Results of QALC system three runs

A first conclusion is that we obtained this year more than a quarter of right answers which represents a quite better score than last year. Even when considering that the question set of this year did not contain complex questions as the definition questions of TREC10.

We evaluated also our three runs thanks to the patterns given by NIST. The number of right answers is given in the first column of Table 3. Since this evaluation takes into account neither the unsupported answers nor the inexact, these results are quite better. Furthermore, it was interesting for us to look at the results given separately by each strategy, that is to say, without processing the final comparison and selection presented in the preceding paragraph. These are given in column 2, which contains the right answers at first rank, while the third column contains the number of right answers at another rank (between second and fifth rank).

Right answers	Right answers at first rank	Right answers at another rank
Run 1 : 152	Nist1: 128	65
Run 2 : 155	Nist2: 132	66
Run 3 : 165	MG: 136	56
	Web: 122	55

Table 3: Right answers at different rank and for different strategies

Comparing the two first columns, we notice that our choice for an architecture maintaining two different search strategies until the final selection, was a good choice. Indeed, it is obvious that returning systematically the first answer of this set is not without any risk: right answers may often be found between second and fifth rank. Even if the two algorithms used in the final selection can yet be improved, we think they are a promising suggestion for the selection step.

6. Conclusion

TREC11 QA track introduced a new evaluation criterion giving even more importance to the reliability of answers. Indeed, weighting answers is always of great consequence because it determines the answers ranking, but it is particularly important in this case. Table 3, in section 5, shows that a largest number of correct answers are found at the top five ranks, and particularly at first rank (from about 66% to 70%). Actually, we made a real endeavour to introduce a number of weighting criteria at three stages of QALC processing: first, weights assigned to selected document sentences, then potential answers weighting during the extraction process, and finally weights assigned to redundancy of answers retrieved from two corpora, TREC and Web. In such a weighting strategy, the difficulty is to balance the relative weights provided by the different criteria. Weighting criteria that we used are from different kind: lexical, syntactical, semantic and statistical. Latter on, additional criteria based on syntactical dependencies will have to be added.

References

Ferret O., Grau B., Hurault-Plantet M., Illouz G. and Jacquemin C. (2001). *Terminological variants for document selection and question/answer matching*, ACL 2001 Workshop on Open-Domain Question Answering, Toulouse, France.

Fillmore C., Baker C., Sato H. (2002). *Seeing arguments through Transparent Structures.* LREC 2002, Las Palmas de Gran Canaria, Spain, pp. 787-791.

Jacquemin, C. (2001). *Spotting and Discovering Terms through NLP*. Cambridge, MA: MIT Press.

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

Soubbotin M.M., Soubbotin S.M. (2001). *Patterns of Potential Answer Expressions as Clues to the Right Answers*. TREC 2001 Notebook, Gaithersburg, USA, pp.175-182.