FRASQUES: A Question Answering system in the EQueR evaluation campaign

Brigitte Grau[*], Anne-Laure Ligozat[*], Isabelle Robba[*], Anne Vilnat[*], Laura Monceaux[†]

[*]LIMSI-CNRS
BP 133
91 403 Orsay Cedex, France
{firstname.lastname}@limsi.fr
[†] LINA
2, rue de la Houssinière
BP 92 208
44322 Nantes Cedex 3, France
Laura.Monceaux@lina.univ-nantes.fr

Abstract

Question-answering (QA) systems aim at providing either a small passage or just the answer to a question in natural language. We have developed several QA systems that work on both English and French. This way, we are able to provide answers to questions given in both languages by searching documents in both languages also. In this article, we present our French monolingual system FRASQUES which participated in the EQueR evaluation campaign of QA systems for French in 2004. First, the QA architecture common to our systems is shown. Then, for every step of the QA process, we consider which steps are language-independent, and for those that are language-dependent, the tools or processes that need to be adapted to switch for one language to another. Finally, our results at EQueR are given and commented; an error analysis is conducted, and the kind of knowledge needed to answer a question is studied.

1. Introduction

The growing interest for searching precise information on the Web or in collections of documents leads many researchers to develop question-answering (QA) systems. Such systems aim at providing either a small passage or just the answer to a question in natural language that pertains to factual information. Typical examples are “When was François Mitterrand elected as president of France for the first time?”, “Which animal becomes white in Winter?”, “What is NATO?”. In order to enlarge the search space, we work on two languages: English and French. This way, we are able to provide answers to questions given in both languages by searching documents in both languages also. At this time, we have developed three systems: QALC only processes English language, FRASQUES only processes French language, and the last one, MUSQAT, answers French questions in English. The three systems have been evaluated in QA evaluation campaigns. QALC participated to TREC[1] evaluation campaigns (from Trec8 to Trec11), MUSQAT participated to QA@CLEF[2] (in 2004 and 2005). We will present here FRASQUES and its results at the EQueR campaign (Ayache et al., 2005), the first campaign for Question Answering systems for French. EQueR is part of a larger evaluation campaign for natural language processing, EVALDA, which itself is part of a Technolangue program supported by the French Research Ministry.

As we first developed QALC, we built FRASQUES by adapting it to French. So, when presenting FRASQUES in this paper, we will present how we modified the different processes that compose it. For this purpose, we will first present FRASQUES architecture in section 2. Then, section 3 will detail the modules, precise which ones are language-independent, and for those that are language-dependent, the changes that have been made to adapt them to a new language. We will finish by presenting and analyzing our results in section 4 before concluding.

2. Architecture of FRASQUES

Answer retrieval can be seen as a process able to recognize affirmative form of questions in texts, and thus to match a question with a sentence or a larger passage of text. The matching is possible only if linguistic variations between the two expressions are taken into account. Answering sentences can be seen as paraphrases of the questions, and thus it is important to recognize every possible variation of the question: synonyms (for example to elect and to vote), morphologic variants (for example to elect and election) and syntactic combinations of them (elected as president and presidential election). Thus, our QA systems focused on the recognition of these variations in the retrieved documents. Fastr (Jacquemin, 1996), a transformational shallow parser for the recognition of term occurrences and variants, will deal with multi-terms variants and extraction patterns with answer paraphrases.

The architecture of FRASQUES (see Figure 1) is classically composed of tree main modules, namely a question analyzer, a document processing module and an answer extraction module. They all communicate by a central XML file. In order to process different languages, we defined common tags for equivalent language-dependent processes: POS tags and Named Entity tags.

Question parsing is done thanks to the XIP parser (Aït-Mokthar et al., 2002), which builds chunks and syntactical dependencies. Then, a set of rules calculates the focus (the object of the question that should be present in the answer) and the expected answer type, which can be either a named entity type or a general type, like colour, metal or music

[1] www.trec.nist.gov

[2] www.clef-qa.itc.it

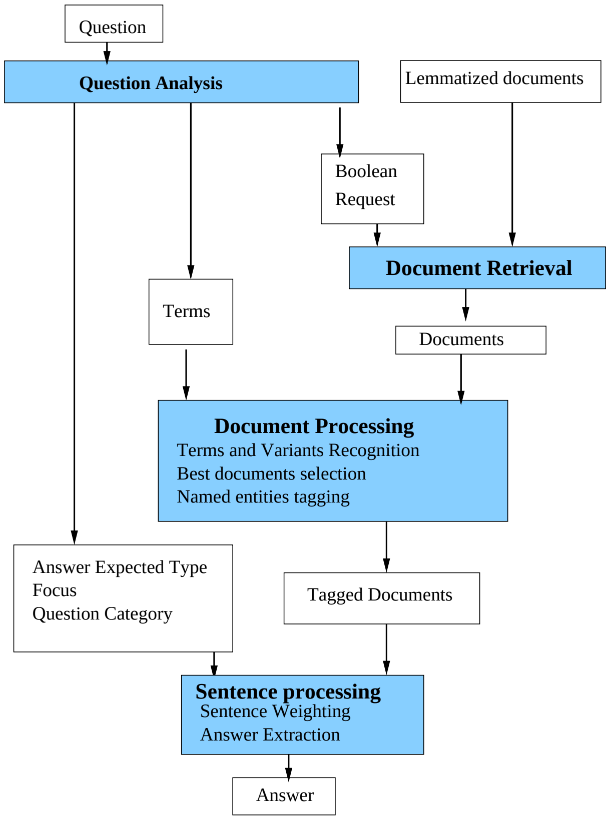

Figure 1: Architecture of FRASQUES, our question answering system

instrument. This latter type is a word of the language. The search engine we use to explore the corpus is Lucene [3], a Boolean search engine. The request sent to Lucene is made with the not empty words of the question, and, if no document is returned, a relaxed request is sent. The retrieved documents are then given to Fastr and the documents are reordered according to the presence of the question terms and their variants. Finally a subset of fifty documents is selected, and these documents are annotated with named entities types. The last module first computes a weight for each sentence of the selected documents, depending on the presence of question terms or variants, and only the sentences with a certain weight are kept. Then, we use a different approach if the expected answer is a named entity or a phrase of a general type. In this second case, we select answers by applying extraction patterns on the candidate sentences. These patterns are based on a morpho-syntactic tagging of the sentences and a particular tagging of the question terms or their synonyms. Patterns are recognized in sentences thanks to the Cass parser (Abney, 1996).

3. Building a multilingual system

3.1. Question analysis

The question analysis module relies on a part-of-speech tagging and a syntactic parsing. Then several elements of information are determined: expected answer type, focus, question category, proper names of the question... The results of the question analysis will be used by all the other modules of the system.

An example of question analysis is given in Figure 2.

Focus: Générale des eaux
Question category: instance
Expected named entity type: ORGANIZATION

Figure 2: *Example of a question analysis:* ***Citez une filiale de la Générale des eaux ?***

In order to be able to switch from English to French, our first step had to consist in adapting the tagger and parser to the new language. The tool we use for the part-of-speech tagging is the TreeTagger [4] in English, and both the TreeTagger and the XIP part-of-speech tagging step in French (the results of each tagger are merged in order to improve the tagging quality). TreeTagger parameter files exist in both English and French, and the output is the same for all the existing languages; XIP's output is converted into the TreeTagger format. As for the syntactic parser, two different tools are used: XIP for French in EQueR, and Cass for English. It was thus decided to project the syntactic analysis of the question in a more neutral format, this of the syntactic parsers evaluation campaign EASY [5]. The output of each parser is converted into this format so that the rest of the question analysis be independent of the used parser.
Then, the morpho-syntactic and syntactic analyzes are used to infer the characteristics of the question. Specific lexicons and rules have been developed for this task, but the core of this module is independent from the language. Thus, both languages are processed in a parallel way and some of the information returned use the same terms in French and in English, like for example the question category or the expected named entity type. Adding another language is therefore made easier, since only the lexicons and the pattern files should change, as summed up in Figure 3. This module was evaluated on corpora of similar questions in French and in English, and its results on both languages are quite close: around 90% of recall and precision for the expected answer type for example (for more details, see (Ligozat et al., 2006)).

3.2. Document retrieval

The search engine for the French collection is Lucene, a Boolean engine. The English collection is searched by MG [6]. As MG can operate Boolean or ranked retrievals, we exploited this characteristic to first conceive Boolean queries, and if they brought out too few or too many documents, we complement them by a ranked retrieval based on the cosine measure. The collection was indexed using the stemming option of MG. Queries are based on POS categories of words and a set of stop words. Proper Nouns and numeric data in the question are favoured.
For the French collection, we switched to Lucene as MG had problems with accented words. As Lucene does not

[3] http://lucene.apache.org/

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[5] http://www.limsi.fr/Recherche/CORVAL/easy/

[6] MG for Managing Gigabytes http://www.cs.mu.oz.au/mg/

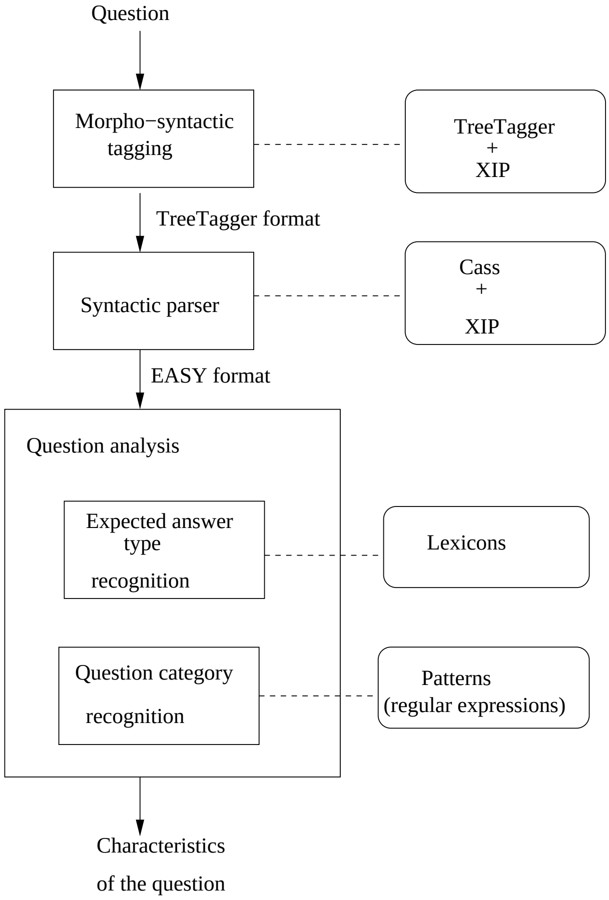

Figure 3: Question analysis module

provide a stemmer, and as all our processes work on lemmatized texts, we decided to tag the whole collection using the XIP lemmatizer, combined with the TreeTagger for some specific cases, before indexing it. At the moment, in French, we only make use of basic Boolean queries, based on the question words and not their synonyms, thus the search engine returned documents containing the exact answer for only 73% to 76% of the questions compared to about 80% in English. Fortunately, the further election of fifty of these documents after re-indexation by Fastr did not entail any further loss.

3.3. Document Processing

Terms (composed of either a single word or of several words) are extracted from the questions through part-of-speech filtering according to patterns describing noun phrases. Then the occurrences or variants of these terms in the documents are detected by Fastr rules enabling morphological and semantic transformations. Morphological families and set of synonyms are data given to Fastr. Thus, dealing with several languages requires conceiving transformational rules in each language.
The following pattern extracts the occurrence *making many automobiles* as a variant of the term *car maker*:

```
VM('maker') RP? PREP?
 (ART (NN|NP)? PREP)?
 ART? (JJ| NN| NP |VBD|VBG)0-3
 NS('car')
```

VM('maker') is any verb in the morphological family of the noun maker and NS('car') is any noun in the semantic family of car. In French the rule for retrieving the variant is the same, while the original term is described by *fabricant de voiture (NN of NN)*.
Named Entities types are classical and we developed two modules for recognizing them. A first one for noun phrases: person, organization, state, city and a second module for numeric entities: different kinds of date (with the year, a day of a month, etc.), time, duration, period, etc. The first system is based on manually developed sets of rules that rely upon lexical information, linguistic constraints of language and contextual information. It was develop for French and English (Elkateb, 2003). Rules combine constraints on words based on word features that are: the original form of the word, its lemma, its POS category, its semantic type(s) if it belongs to lexicon(s) (for instance first name or city) and its typography. A high level language was conceived for these rules and allows the description of all kinds of entities with few rules. For English, there are 7 rules for person recognition, 9 for locations and 7 for organizations. In French, there were 11 rules for persons, 19 for locations and 13 for organizations.
Here an example of a complex rule that tags an enumeration of persons in the input text. An enumeration is a sequence of persons names separated with commas. We have to tag each person alone.

```
Pattern: PERSON_COMMA
({token == ","}) /*UNTAG*/
({lookup IN "firstname"} OR
    {typography == "PMAJ"}) [1,5]
({typography == "ROM"} [0,1]

Rule: ENUMERATE_PERSONS
({lookup IN "firstname"} OR
    {typography == "PMAJ"}) [1,5]
({typography == "ROM"} [0,1]
(PERSON_COMMA) [0, 10]
-->
{Type ="PERSON", Rule = "Enumeration"}
```

This rule is the same in the two languages. PMAJ signifies that the first letter is capitalized, ROM that the word has to be in Roman characters. Only the resources *firstname* has to be adapt for each language.
The second module is developed in Perl language and is based on rewriting rules, also manually developed, but they are more numerous than those of the first system. Thus, only rules and data were to be changed, the core system remaining the same. For numeric entities, we based our conception on the English rules, and adapted them to French. Thus, adapting each rule written for English was faster than re-conceiving the initial set of rules, as for those local grammars, English and French are very close languages.

3.4. Sentence Processing

3.4.1. Patterns

The answer extraction module aims at extracting short answers from the candidate sentences. Candidate sentences are themselves extracted from the selected documents and

Level 1: Nominal Groups
NGFoc -> DT? RB? (ADJ (CC ADJ)?)? FC RB? ;
NG -> DT? RB* ADJ* (NN|NNS)+ RB* ADJ* ;
NGNP -> DT? RB* ADJ* (NP|NPS)+ ;
Level 2: Answer tagging
FocusLeftAnswer -> b= (NG|NGNP) SEP NGFoc ;

Figure 4: *Pattern example in Cass grammar rule formalism*

then reordered by Fastr. In Frasques two strategies are adopted depending on the type of the expected answer. If the expected answer is a named entity, Frasques returns the named entity which is the nearest of the barycenter of the question words or of their variants. In the other cases, the extraction is more difficult, because we cannot rely as strongly on the type of the answer, which can be as well a verb a noun or an adjective, and the answers cannot be tagged as for named entities. Hence, we apply different patterns according to the type of the question and taking into account information provided by the question analysis.
In the EQueR evaluation, we chose to use Cass parser for this extraction and we wrote patterns using Cass grammar rule formalism. Here, Cass is not used as a sentence parser, but its rule formalism enables us to tag the possible answers in the candidate sentences.
Cass parser takes two files as input: in the first file, the sentences to parse are tagged by a morpho-syntactic parser (the TreeTagger is used in our architecture); and in the second file, the main characteristics of the question (provided by the question analysis) are tagged with particular tags: the focus (FC), the main verb (MV), the expected answer general type (GT). When they are available, the synonyms of these characteristics are also tagged.
Cass is a cascaded finite-state parser; its rules are organised in several different levels and written as regular expressions. Figure 4 shows how we use these levels to tag the possible answers. The first level enables us to tag the different nominal groups (NG): NGFoc for instance is a nominal group that contains the question focus, while NGNP is a nominal group containing a proper noun (NP). Then, at the second level, the pattern *FocusLeftAnswer* expresses that if the candidate sentence contains an NGFoc followed by a separator followed by an NG then the NG is tagged as a possible answer.

These patterns enable us to correctly answer to the question *Quelle est la monnaie nationale en Hongrie ?* (*What is the national currency in Hungary?*). In this question the focus is *monnaie* (*currency*) and the main verb is *être* (*to be*). The correct answer is in the following candidate sentence *La Banque nationale de Hongrie a dévalué la monnaie nationale, le forint, de 1 %...* (*The Hungarian National Bank devaluated the national currency, the forint, of 1%...*). The final extraction module returns the NG *the forint* tagged by *b=*:

```
[FocusLeftAnswer
 [NGFoc
   [DT the]
   [ADJ national]
   [FC currency]]
 [VIRG ,]
 b=[NG
     [DT the]
     [NN forint]]
      ]
```

Using Cass formalism to write the extraction patterns presents several advantages. First of all, the rules are easily and quickly written. Moreover Cass application is not time consuming: as described in (Abney, 1996), the speed of Cass is quite sensitive to the number of levels in the grammar, and in our patterns the maximum number of level is 4. Lastly, the passage from French to English is not difficult, few patterns are added, most of them consist in a simple translation of the Tree Tagger tags.

3.4.2. Sentence weighting and answer extraction

Sentence weighting is based on the presence rate of the question terms, either in their question form or in a variant form. The words are weighted according to a general resources built on a large corpus that encodes the significativity rate of a word according to its total number of occurrences in the corpus and the number of documents that contain it. A reward is added if some of the following features are present: words in the candidate sentence exactly the same as in the question, proximity of the words, presence of a named entity of the expected type. In order to extract the answer string, sentences are analyzed with Cass for applying extraction patterns. Patterns are language-dependent; however the process that tags the words depending of their role in the question and extract the phrases that are potential answers are quite the same in the two languages.

4. Results analysis

4.1. EQueR results

The EQueR campaign aimed at offering a evaluation opportunity for QA systems for French. The document collection came from different sources such as the newspapers *Le Monde*, or *Le Monde Diplomatique* or Senate debates. The types of questions proposed were quite varied: factual questions, definition questions, list questions (such as *Quelles sont les 4 religions pratiquées en Hongrie ?*), yes/no questions, as well as reformulations of some of the factual questions. Some of the questions had no answer in the corpus; the expected answer was then *NIL*.
The general task (there was also a medical task) contained 500 questions with 407 factual questions, 32 definition ones, 31 list ones, and 30 yes/no ones. For each question, the systems were expected to return either the short answer to the question or a passage containing it. Up to 5 answers were authorized.
For this campaign, we submitted two runs, in order to test different selection document strategies. For the first run, all proper names were used without considering the threshold of 200 documents; for the second run, we checked the number of documents after each query.
Table 1 presents the results we obtained at EQueR, and compares them to those obtained at TREC11 (for English).

	Long answers	MRR	Short answers	MRR
EQueR run1	42%	0.37	26%	0.22
EQueR run1 (corrected)	60%	0.48		
EQueR run2	37%	0.32	24%	0.20
Trec9	56%	0.40		
Trec11 (1st rank)			28%	

Table 1: EQueR and TREC results

In the runs given for the campaign, a error remained in our last module, which reduced the size of the long answers, so we indicated here what our results could have been without this error. It is interesting to notice that the results on short answers are not far from those obtained by our English system QALC in the Trec evaluations. EQueR results being for the first five ranks, while Trec ones being for the first rank only, they could have been expected to be higher. Yet, in Trec, the Web was used as an additional source of information, and the answers provided by the collection and the Web were merged in order to improve the performance of our system.

4.2. Importance of synonyms

We conducted a first study that measures the rate of question words and synonyms in the answering sentences of our system as well as of the participants in EQueR (Barbier et al., 2005). The goal of this study was to determine what type of variants from the question words was most useful when searching for the answer.

First, we checked that synonyms could be found for EQueR questions. While the average number of words per question was of about 6, around 13 Fastr synonyms and 7 EuroWordNet synonyms could be found. Then we studied the corpus formed by the correct passages given by the participants and we calculated the presence rate of synonyms in these passages. Among these passages, 82% do not contain any Fastr synonym, and 88% do not contain any EuroWordNet synonym. The question words are much more frequent (60%) in the passages than the Fastr (4%) or EuroWordNet synonyms (3%).

Several reasons explain those rather low rates of synonyms in the corpus. First, the synonym bases are not the ones the other participants use, moreover few of them take into account such knowledge. Second, in EQueR, a lot of correct answers could be found with the words of the question. It seems (it is also true in TREC campaigns) that there is often at least one formulation close to the question, which is probably due to the large amount of documents (1.5 gigabytes). Yet, since very few synonyms of the question words are found in the answers, the issue of the type of knowledge to use in QA systems should be explored.

4.3. Error analysis

In order to be able to improve our system, we traced part of its errors: as our goal was to study the answer extraction, we particularly focused on the questions for which our system extracted incorrect answers from sentences containing good answers. The corpus was thus of 74 questions. Several causes of errors were observed:

- 25 incorrect answers came from the use of a wrong pattern.
- In 7 questions, the question analysis module wrongly returned a named entity as the expected answer type.
- In 11 questions, the question analysis module failed to recognize a named entity type: for the question, *À quel endroit s'est terminée la 21e édition du Dakar ?* (In what place did the 21st Dakar rally finish?), the question analysis did not return the expected type *LOCATION*.
- In 15 questions, the answer extraction module had to chose between several named entities of the expected type, and chose an incorrect answer: for the question *Combien y a-t-il de membres non-permanents au Conseil de Sécurité de l'ONU ?* (How many non-permanent members are there at the UN Security Council?), the system returns the incorrect answer *cinq* present in the sentence *Le Conseil de scurit compte actuellement quinze membres dont cinq permanents disposant du droit de veto (...) et dix non-permanents.*
- In 11 questions, the correct answer was not tagged with the correct named entity type: for the question *Où sont exposées "Les Noces de Cana" ?* (Where are "The Wedding Feast at Cana" displayed?), the answer *Louvre* is tagged as a proper name and not a location.
- The remaining 5 errors are diverse.

This error analysis was useful since it helped us spot the weaknesses of our system, and we conducted more thorough studies for some of the above issues: the pattern sets have been revised, and the lexicons used by the question analysis module were extended.

5. Conclusion

In this article, we have presented our French Question Answering system FRASQUES. This system has been adapted from our English Question Answering system QALC, which has been evaluated in the TREC campaign. All along this paper, we detailed the processes which are language independant, and for those that are language-dependent, we present the changes that have been made to adapt them to a new language. Our system has been built to allow a rather simple shift from a language to another. We then develop a system which is able to provide answers to questions given in both languages by searching documents in both languages also, that is MUSQAT which participates to multilingual evaluations (CLEF campaign). With these three systems, we are able to compare the results we obtained during the different campaigns. We also try to obtain benefits of these different experiments, to enhance our global results.

6. References

Steven Abney. 1996. Partial parsing via finite-state cascades. In */ESSLLI '96 Robust Parsing Workshop/*.

Salah Aït-Mokthar, Jean-Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: incremental deep parsing. *Journal of Natural Language Engineering*, 8(3-2).

Christelle Ayache, Brigitte Grau, and Anne Vilnat. 2005. Campagne d'évaluation equer-evalda : évaluation en question-réponse. In *Working Notes, CLEF Cross-Language Evaluation Forum*, Vienna, Austria.

Vincent Barbier, Brigitte Grau, Anne-Laure Ligozat, Isabelle Robba, and Anne Vilnat. 2005. Semantic knowledge in question-answering systems. In *IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, Edinburgh, UK.

Faïza Elkateb. 2003. Extraction d'entités nommées pour la recherche d'informations précises. In *4ème congrès ISKO-France, L'organisation des connaissances*, Grenoble, France.

Christian Jacquemin. 1996. A symbolic and surgical acquisition of terms through variation. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 425–438.

Anne-Laure Ligozat, Brigitte Grau, Isabelle Robba, and Anne Vilnat. 2006. L'extraction des réponses dans un système de question-réponse. In *Traitement Automatique des Langues Naturelles (TALN 2006)*, Leuven, Belgium.