FIDJI: Web Question-Answering at Quaero 2009

Xavier Tannier, Véronique Moriceau

LIMSI-CNRS
University Paris-Sud 11
91403 Orsay, France
firstname.surname@limsi.fr

Abstract

This paper presents the participation of FIDJI system to the Web Question-Answering evaluation campaign organized by Quaero in 2009. FIDJI is an open-domain question-answering system which combines syntactic information with traditional QA techniques such as named entity recognition and term weighting in order to validate answers through multiple documents. It was originally designed to process "clean" document collections. Overall results are significantly lower than in traditional campaigns but results (for French evaluation) are quite good compared to other state-of-the-art systems. They show that a syntax-based strategy, applied on uncleaned Web data, can still obtain good results. Moreover, we obtain much higher scores on "complex" questions, *i.e.* 'how' and 'why' questions, which are more representative of real user needs. These results show that questioning the Web with advanced linguistic techniques can be done without heavy pre-processing and with results that come near to best systems that use strong resources and large structured indexes.

1. Introduction

FIDJI (Finding In Documents Justifications and Inferences) is an open-domain question-answering (QA) system for French (Moriceau et al., 2009). It combines syntactic information with traditional QA techniques such as named entity recognition and term weighting in order to validate answers through different documents.
We present in this paper the results obtained by FIDJI at French Quaero 2009 evaluation. *Quaero*[1] is a program promoting research and industrial innovation on technologies for automatic analysis and classification of multimedia and multilingual documents. Among the many research areas concerned by Quaero, a yearly evaluation campaign of question-answering systems has been organised in both French and English languages (Quintard, 2008; Quintard, 2009; Quintard et al., 2010).

One of the goals is to evaluate the capacity of a system to answer user questions of different kinds within a raw Web corpus. QA Quaero campaigns have the following specificities:

- A 2 million Web page corpus, collected without quality filtering by Exalead[2]. This means that any kinds of Web pages (blogs, forums, spam, news, institutions, etc.) can be found, as well as some non-French pages. We obviously expect these documents to be generally much less respectful of French syntactic standards than traditional newspaper articles.
- 500 questions are elaborated by an independent partner, without looking at the corpus, but from search engine logs. Among them, it turned out that 412 had an answer in the corpus (answers found by the systems or the assessor). Results presented in this section consider only these 412 questions.
- In 2009, a specific effort has been devoted to complex questions ('how' and 'why' questions), that are usually not very studied in question-answering. 102 questions (out of 500) are complex.

In this article, the system FIDJI is briefly presented, examples of analyses are given, and results obtained at Quaero 2009 are detailed.

2. FIDJI

The objective is to produce answers which are fully validated by a supporting text (or passage) with respect to a given question. The main difficulty is that an answer (or some pieces of information composing an answer) may be validated by several documents. For example:

> Question: *Which French Prime Minister committed suicide?*
> Answer: Pierre Bérégovoy
> Passage 1: The French Prime Minister Pierre Bérégovoy warned Mr. Clinton against...
> Passage 2: Two years later, Pierre Bérégovoy committed suicide after he was indirectly implicated...

In this example, the information "*French Prime Minister*" and "*committed suicide*" are validated by two different passages. None of the passages could validate the answer by it-self. This is made possible by the possibility to decompose the question into two sub-questions, *e.g.* "*Who committed suicide?*" and "*Are they French Prime Minister?*".

Syntactic analysis can provide with these kinds of accurate decompositions. Almost all recent researches are based on a syntactic and semantic analysis and often imply a pre-processing of the whole document collection (Katz et al., 2005; Hartrumpf et al., 2008; Sun et al., 2005).
Our aim is to extract and validate answers by going beyond the exact syntactic matching between questions and answers. This must be done without using any semantic

[1] http://www.quaero.org
[2] http://www.exalead.com

resources and with as less pre-processing as possible: this is a necessary condition if the system works on large collections such as the Web.

In this context of answer validation, the strategy to apply (validation through one or several documents) can be guided by the question, and especially by the expected answer type. Indeed, a lot of factoid questions expect an answer of a specified type. This type can be:

- A named entity type as in "*Who is the president of France*" which expects an answer of type person;
- A more specific type as in "*Which Russian president attended the G7 meeting in 2007?"* which also expects an answer of type person, but the type is here explicitly specified in the question (Russian president).

Our approach consists in checking if all the characteristics of a question (namely the dependency relations and the answer type) may be retrieved in one or several documents. In this context, FIDJI has to detect syntactic implications between questions and passages containing the answers and to validate the type of the potential answer in this passage or in another document. Our system relies on syntactic analysis provided by XIP (Aït-Mokhtar et al., 2002), which is used to parse both the questions and the documents from which answers are extracted.

Figure 1 presents the architecture of FIDJI. The document collection is indexed by the search engine Lucene[3]. First, the system submits the keywords of the question to Lucene: the first 100 documents are then processed (syntactic analysis and named entity tagging). Among these documents, FIDJI looks for sentences containing the highest number of syntactic relations of the question. Finally, answers are extracted from these sentences and the answer type, when specified in the question, is validated (Moriceau and Tannier, 2009). The following examples illustrate how FIDJI extracts answers.

2.1. Example 1

Question analysis provides dependency relations, the question type and the expected answer type. For example:

Question: Quel premier ministre s'est suicidé en 1993 ?
(*Which Prime Minister committed suicide in 1993?*)
Dependencies: `DATE(1993)`
`PERSON(ANSWER)`
`SUBJ(se suicider, ANSWER)`
`attribut(ANSWER, ministre)`
`attribut(ministre, premier)`
Question type: factoid
Expected answer type: person (specific answer type: prime minister)

The question is turned into a declarative sentence where the answer is represented by the 'ANSWER' lemma. The following sentence is selected because it contains the highest number of dependency relations:

Pierre Bérégovoy s'est suicidé en 1993.
(*Pierre Bérégovoy committed suicide in 1993.*)
Dependencies:
`DATE(1993)`
`PERSON(Pierre Bérégovoy)`
`SUBJ(se suicider, Pierre Bérégovoy)`

Pierre Bérégovoy instantiates the ANSWER slot of the question dependencies and becomes a candidate answer. The named entity type (person) and the first three dependencies of the question are validated in this sentence. In order to fully validate the candidate answer, the system searches the missing dependencies (`attribut(Pierre Bérégovoy, ministre)` and `attribut(ministre, premier)`) in a single sentence of the whole document collection. These dependencies will be found in any sentence speaking about "*le premier ministre Pierre Bérégovoy*" (*Prime Minister Pierre Bérégovoy*) and the answer will be validated.

2.2. Example 2

For complex questions, it is obvious that answers are not always short phrases. For this reason, FIDJI provides a full passage as an answer. On these kinds of questions, the system behaves as a classical passage retrieval system, except that candidate passages are retrieved through syntactic relations and relevant discourse markers (about 100 nouns, verbs, prepositions and adjectives) instead of keywords only. Here is an example of a complex question.

Question: Pourquoi le ciel est-il bleu ?
(*Why is the sky blue?*)
Dependencies: `attribut(ciel, bleu)`
Question type: complex (why)
Expected answer type: ∅

The following passage is selected because it contains all the dependency relations of the question and a causal marker:

Et si le ciel est bleu, c'est ***à cause de*** *la diffusion de Rayleigh qui est la plus importante dans le bleu (ondes électromagnétiques, ...).*
(And if the sky is blue, it is ***because of****...)*
`attribut(ciel, bleu)`
`VMOD(être, diffusion)`
`PREPOBJ(diffusion, à cause de)`
`...`

FIDJI was originally designed to process "clean" document collections (such as CLEF collections composed of well-formed and syntactically correct news articles) and obtains good results (66% of correct answers on CLEF 2005) (Moriceau et al., 2009). However, the Quaero campaign represents a good opportunity to show that linguistic-oriented techniques can be applied to large, uncontrolled collections.

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

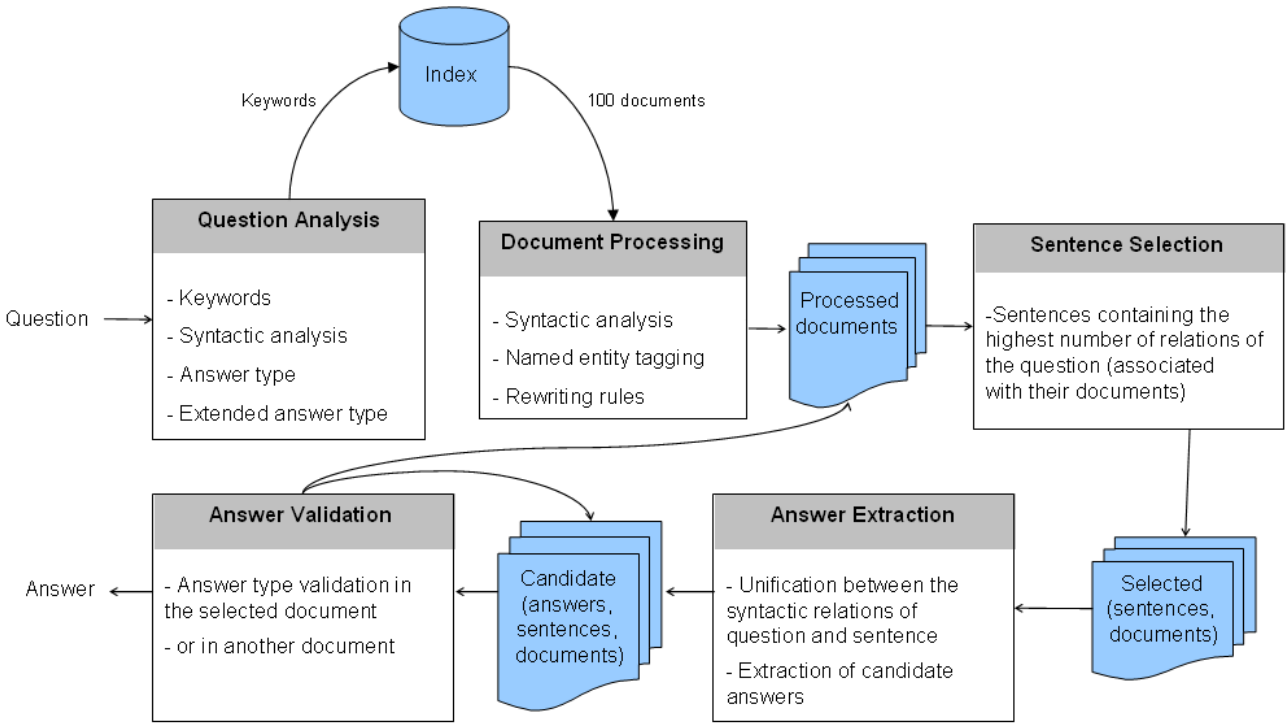

Figure 1: FIDJI architecture

In the context of Quaero, FIDJI has been designed in order to build a realistic question-answering system on the Web. For that purpose, it avoids heavy linguistic pre-processing, that makes currently impossible any scaling to a very large collection of texts. Only a traditional bag-of-word indexing is necessary, and all fine linguistic analysis is performed online on a small subset of documents. Also, no large knowledge base has been used, so that the system is easier to maintain and adaptable to various languages.

3. Results at Quaero 2009 campaign

As described in Section 1., specificities of Quaero QA evaluation campaign are, among others, new questions types (boolean, complex questions) and a collection of documents collected from the Web. This last point leads to many problems, among which:

- Web pages contain structured information (as tables, titles) that can contain very useful information and are not handled by participating QA systems.
- No spam filter has been applied to the corpus, and many questions, especially those concerning known people or events, are polluted by these spam documents.
- About 10% of the documents (according to an estimation completed by Exalead, personal communication) are not in the appropriate language (*i.e.* French for French evaluation, English for English evaluation).
- Many HTML pages converted from PDF or RTF documents had conversion format problems with special characters, leading to split words.

Figure 2 shows two examples of problems raised by Web documents. The first document is a spam, containing the same keywords ("jeu ligne enfants internet") many times. These keywords are between <b> HTML tags and are hidden by the CSS when shown to the reader. However, they are extracted by the HTML-to-text extractor, what makes the text impossible to analyse usefully by the system.

The second document is a form for job add consultation. It contains selection lists with all regions and departments of France. The content of these lists is also extracted to the plain text version, while the regions of France are not relevant to the subject of the document.

These specificities explain, at least partially, that overall results are significantly lower than in traditional campaigns (for the same systems). This is the case for all participants, that are recognized as the best systems for French.

For each question, participants may return up to three answers. Each answer is a combination of a short string and the passage which supports it. More details concerning Quaero guidelines can be found in (Quintard et al., 2010). Overall results are presented in Table 1, with Mean Reciprocal Rank, first hit success (relevance) and hit success until rank 3 (accuracy) displayed for correct short answer only, correct support text only or both.

Table 2 presents results for short answer by question types, using the same measures. Best system scores are also presented, or second best when FIDJI ranked 1st. Even if FIDJI still gets lower scores than best systems on traditional factual questions, results on complex questions are much higher. This is very promising since these types of questions are representative of real user needs, and we

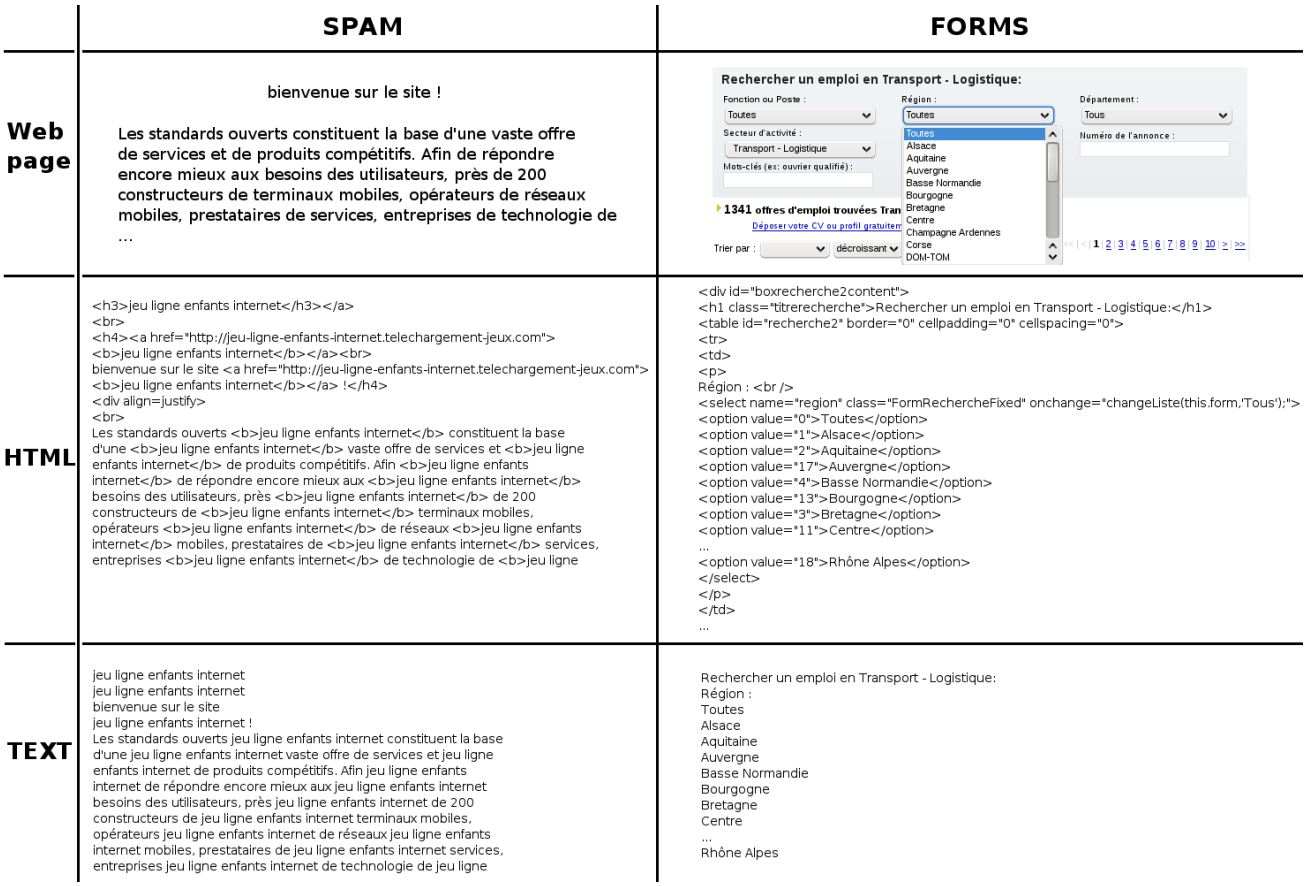

Figure 2: Examples of problems raised by Web pages

Answer	Metric	FIDJI	Best system
Short	MRR	39.14	43.33
	Relevance	34.61	40.51
	Accuracy	45.12	46.41
Short + support	MRR	37.73	41.15
	Relevance	33.33	38.20
	Accuracy	43.58	44.10
Support	MRR	46.66	48.54
	Relevance	42.82	44.61
	Accuracy	51.79	53.33

Table 1: Overall results, Quaero 2009, French.

Type	Metric	FIDJI	Best system (or 2nd best)
Complex “how”	MRR	**49.01**	14.70
	Relevance	**41.17**	11.76
	Accuracy	**58.82**	17.64
Complex “why”	MRR	**31.74**	19.04
	Relevance	**23.80**	19.04
	Accuracy	**42.85**	19.04
Factual	MRR	37.18	**54.00**
	Relevance	32.99	**50.21**
	Accuracy	42.85	**57.93**
Definition	MRR	**26.11**	16.66
	Relevance	**20.00**	16.66
	Accuracy	**33.33**	16.66
Boolean	MRR	60.71	**82.75**
	Relevance	60.71	**82.75**
	Accuracy	60.71	**82.75**

Table 2: Results by question type (short answer).

proved that specific efforts on these aspects can lead to interesting results.

These results show that questioning the Web with advanced linguistic techniques can be done without heavy pre-processing and with results that come near to best systems that use strong resources and large structured indexes.

4. Conclusion

This article presented the results obtained by the system FIDJI during the Quaero QA evaluation campaign for French in 2009. FIDJI has been designed in order to build a high level, realistic question-answering system on the Web. For that purpose, it avoids heavy linguistic pre-processing and knowledge bases. Despite of this, the system ranks very close to the best state-of-the-art system. In order to make the system more robust, we plan to work on the text extraction from the Web pages. Resolving some document issues described in previous section should lead to a better precision.

Acknowledgement

This work has been partially financed by OSEO under the Quaero program.

5. References

S. Aït-Mokhtar, J.P. Chanod, and C. Roux. 2002. Robustness beyond shallowness: Incremental deep parsing. *Natural Language Engineering*, 8:121–144.

S. Hartrumpf, I. Glöckner, and J. Leveling. 2008. University of Hagen at QA@CLEF 2008: Efficient Question Answering with Question Decomposition and Multiple Answer Streams. In P. Forner, A. Peñas, I. Alegria, C. Forascu, N. Moreau, P. Osenova, P. Prokopidis, P. Rocha, B. Sacaleanu, R. Sutcliffe, and E. Tjong Kim Sang, editors, *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

B. Katz, G. Borchardt, and S. Felshin. 2005. Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, Pittsburgh.

V. Moriceau and X. Tannier. 2009. Apport de la syntaxe dans un système de question-réponse : étude du système fidji. In *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2009)*, Senlis, France.

V. Moriceau, X. Tannier, and B. Grau. 2009. Utilisation de la syntaxe pour valider les réponses à des questions par plusieurs documents. In *Proceedings of COnférence en Recherche d'Information et Applications, CORIA*, Presqu'île de Giens, France.

L. Quintard, O. Galibert, G. Adda, B.Grau, D. Laurent, V. Moriceau, S. Rosset, X. Tannier, and A. Vilnat. 2010. Question Answering on web data: the QA evaluation in Quaero. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC'2010)*, La Valette, Malta.

L. Quintard. 2008. Overview of the Quaero 2008 Monolingual Question Answering Track. Laboratoire National de Métrologie et d'Essais. http://www.lne.eu/publications_en/research/quaero-QA-2008-overview.pdf.

L. Quintard. 2009. P2 Evaluation Report: Evaluation Design for Task 3.5 on Question Answering systems. Technical Report Internal Deliverable ID.CTC.12.S3.5.P2, Quaero program, CTC project.

R. Sun, J. Jiang, Y.F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. 2005. Using syntactic and semantic relation analysis in question answering. In *The Fourteenth Text REtrieval Conference (TREC)*, Gaithersburg, USA.