FIDJI in ResPubliQA 2009

Xavier Tannier
CNRS-LIMSI
University Paris-Sud 11
xtannier@limsi.fr

Véronique Moriceau
CNRS-LIMSI
University Paris-Sud 11
moriceau@limsi.fr

Abstract

This paper presents FIDJI results in ResPubliQA 2009. FIDJI (Finding In Documents Justifications and Inferences) is an open-domain question-answering system for French. The main goal is to validate answers by checking that all the information given in the question are retrieved in the supporting texts.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Managment**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Questions beyond factoids

1 Introduction

This paper presents FIDJI's results in ResPubliQA 2009 for French. In this task, systems receive 500 independent questions in natural language as input, and must return one paragraph containing the answer from the document collection. No exact answer is required neither multiple responses. The document collection is JRC-Acquis about EU documentation.

2 FIDJI

FIDJI[1] (Finding In Documents Justifications and Inferences) is an open-domain question-answering system for French. The main goal is to validate answers by checking that all the information given in the question are retrieved in the supporting texts. Our answer validation approach assumes that the different entities of the question can be retrieved, properly connected, either in a sentence, in a passage or in multiple documents. We designed the system so that no particular linguistic-oriented pre-processing is needed.

The document collection is indexed by the search engine Lucene[2] [2]. First, the system submits the keywords of the question to Lucene: the first 100 documents are then processed (syntactic

[1] This work has been partially financed by OSEO under the Quaero program.
[2] http://lucene.apache.org/java/docs/index.html

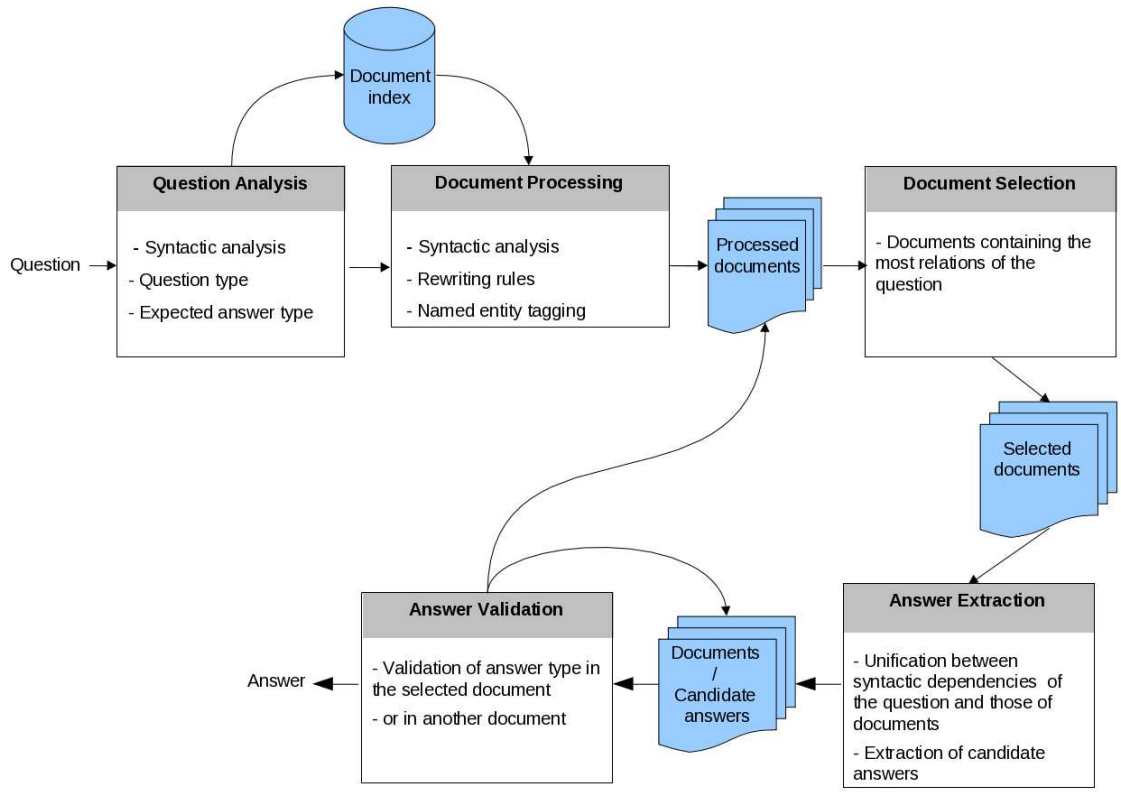

Figure 1: Architecture of FIDJI

analysis and named entity tagging). Among these documents, FIDJI looks for sentences containing the most syntactic relations of the question. Finally, answers are extracted from these sentences and the answer type, when specified in the question, is validated. Figure 1 presents the architecture of FIDJI and more details can be found in [4, 3]. Next sections summarize the way FIDJI extract answers and focuse on ResPubliQA specificities.

2.1 Syntactic analysis

FIDJI has to detect syntactic implications between questions and passages containing the answers. Our system relies on syntactic analysis provided by XIP, which is used to parse both the questions and the documents from which answers are extracted.

XIP [1] is a robust parser for French and English which provides **dependency relations** and named entity recognition. The dependency relations provided by XIP which are used by FIDJI are mainly: SUBJ (subject), OBJ (object), PREPOBJ (prepositional group), NMOD (noun modifier), VMOD (verb modifier), COORDITEMS (coordinated elements) CONNECT (connector introducing clause).

The **named entities** (NE) are tagged using a set of 8 types: person, organization, location, date (defined by XIP), as well as nationality, number, duration, age (that we added). XIP's lieu (location) can be made more specific (country, region, continent...). We also added features to allow for more precise types. For example, for number, we added the following features: length, speed, weight, money, physics, so that "0.55 euro" in "a French stamp costs 0.55 euro" can be tagged as a NE and extracted as an answer to "What is the price of a French stamp?". Other elements are also tagged, as names introducing persons: functions (leader...), professions (minis-

ter...), family indications (father...).

Question analysis consists in identifying:

- The syntactic dependencies given by XIP;
- The keywords submitted to Lucene (words tagged as noun, verb adjective or adverb by XIP);
- The question type:
  - Factoid (concerning a fact, typically who, when, where questions),
  - Definition (What is...),
  - Boolean (expecting a yes/no answer),
  - List (expecting an answer composed of a list of items),
  - Complex questions (why and how questions).
- The expected type(s): NE type and/or (specific) answer type.

The answer to be extracted is represented by a variable (ANSWER) introduced in the dependency relations. The slot noted 'ANSWER' is expected to be instantiated by a word, argument of some dependencies of the parsed sentences. This word represents the answer to the question (see Section 2.2). The question type is mainly determined on the basis of the dependency relations given by the parser. For example:

0015 - *Entre quels pays a été conclu l'accord-cadre de coopération commerciale et économique du 2 avril 1990 ?*
(*Between which countries is the Framework Agreement for trade and economic cooperation of 2 April 1990?*)

- Syntactic dependencies and NE tagging:
  `ATTRIBUTADJ(coopération, commercial)` `ATTRIBUTADJ(coopération, économique)`
  `ATTRIBUT_DE(accord-cadre, coopération)` `VMOD(conclure, ANSWER)`
  `PREPOBJ(ANSWER, entre)` `ATTRIBUT(conclure, accord-cadre)`
  `DATE(2 avril 1990)` `LIEU[PAYS](ANSWER)`
- Question type: list
- Expected type: location (state)

0021 - *Comment encourage-t-on la production de graines de vers à soie ?*
(*How is interest in producing silkworm eggs increased?*)

- Syntactic dependencies and NE tagging:
  `ATTRIBUT_DE(graine, vers)` `ATTRIBUT_DE(production, graine)`
  `DEEPOBJ(encourager, production)` `NMOD(vers, soie)`
  `TOPIC(encourager)`
- Question type: complex
- Expected type: $\emptyset$

2.2 Extracting candidate paragraphs

ResPubliQA answer format is different from traditional QA campaigns. First, answers are not focused, short parts of texts, but full paragraphs that must contain the answer. Second, passages are not indefinite parts of texts of limited length; they must be predefined paragraphs identified in the collection by XML tags <p>.

Although answers to submit to the campaign are full paragraphs, our system is designed to hunt down short answers. For most questions, typically factoid questions, it is still relevant to find short answers, and then to return a paragraph containing the best answer. This is not the case of 'how' or 'why' questions, where no short answer may be retrieved.

FIDJI usually works at sentence level. For the aim of ResPubliQA specific rules, we chose to work at paragraph level. This consisted in specifying that sentence separators were <p> XML tags in the collection, rather than usual end-of-sentence markers.

Once candidate documents are selected by the search engine and analyzed by the parser, the system compares the document paragraphs with question analysis, in order to:

- Extract candidate answers or select a relevant paragraph;
- Give a score to each answer, so that final answers can be ranked.

2.2.1 Factoid questions

Within selected documents, candidate paragraphs are those containing the most dependencies from the question. Once these paragraphs are selected, two cases can occur:

1. Question dependencies with an 'ANSWER' slot are found in the sentence. In this case, the lemma instantiating this slot is the head of the answer. The full answer is composed of the head and its basic modifiers (for a noun phrase: noun complements, adjectives, determiners and coordinated elements; for a verbal phrase: verb complements, subject and object). The eventual NE type and answer type of this answer are checked. Answer type can be validated by different syntactic relations in the text: definition ("The French Prime minister, Pierre Bérégovoy"), attributNN ("Pierre Bérégovoy is the French Prime minister"), and sometimes attribut_de ("la maladie de Parkinson", Parkinson's desease, literally "the disease of Parkinson").
2. The 'ANSWER' slot does not unify with any word of the passage. In this case, the elements having an appropriate NE type and/or answer type are selected in the sentence. This is done in order to counterbalance the many parsing errors (or paraphrases). Often, the sentence contains the answer but syntactic dependencies alone do not lead to it.

If no possible short answer is found, the paragraph is still considered as a candidate answer. But in any case, a paragraph containing an extracted short answer will be prefered if it exists.

Example 1.

0015 - *Entre quels pays a été conclu l'accord-cadre de coopération commerciale et économique du 2 avril 1990 ?*
(*Between which countries is the Framework Agreement for trade and economic cooperation of 2 April 1990?*)

- Syntactic dependencies and NE tagging:

ATTRIBUTADJ(coopération, commercial)	ATTRIBUTADJ(coopération, économique)
ATTRIBUT_DE(accord-cadre, coopération)	ATTRIBUT(conclure, accord-cadre)
VMOD(conclure, **ANSWER**)	PREPOBJ(**ANSWER**, entre)
DATE[DATEABS](2 avril 1990)	LIEU[PAYS](ANSWER)

- Question type: **list**

- Expected type: **location (state)**

The following passage is selected because it contains the dependencies of the question:

Passage: *un accord-cadre de coopération commerciale et économique entre la Communauté économique européenne et la République argentine (3) a été conclu le 2 avril 1990 ;*
(*Considering the Framework Agreement for trade and economic cooperation between the European Economic Community and the Argentine Republic of 2 April 1990;*)

```
ATTRIBUTADJ(coopération, commercial)     ATTRIBUTADJ(coopération, économique)
ATTRIBUT_DE(accord-cadre, coopération)   ATTRIBUT(conclure, accord-cadre)
NMOD(coopération, communauté économique européen)
PREPOBJ(communauté économique européen, entre)
COORDITEMS(communauté économique européen, république argentin)
LIEU[PAYS ](république argentin)
DATE(2 avril 1990)
ORG(communauté économique européen)
```

The slot 'ANSWER' is instantiated by *communauté économique européenne*. As the question type is 'list', the elements of the list has to be found in a 'COORDITEMS' dependency: so, the answers are *communauté économique européenne* and *république argentine*. Finally, the expected answer type is validated: the selected answer is tagged as a location (state).

Example 2.

0026 - *Quel est le nom de la monnaie des états membres depuis le 1er janvier 1999 ?*
(*What is the name of the member states' currency from 1 January 1999?*)

- Syntactic dependencies and NE tagging:
  ```
  ATTRIBUT_DE(monnaie, état)          NMOD(état, membre)
  PREPOBJ(1er janvier 1999, depuis)   DEFINITION(ANSWER, monnaie)
  DATE(1er janvier 1999)
  ```
- Question type: **definition**
- Expected type: $\emptyset$

The following passage is selected because it contains all the dependencies of the question:

Passage: *considérant que le règlement (CE) n 974/98 du Conseil du 3 mai 1998 concernant l'introduction de l'euro (3) prévoit à son article 2 que, à compter du 1er janvier 1999, la monnaie des États membres participants est l'euro ;*
(*Whereas Council Regulation (EC) No 974/98 of 3 May 1998 on the introduction of the euro (3), provides in Article 2 that from 1 January 1999 the currency of the participating Member States shall be the euro*)

```
ATTRIBUTADJ(membre, participant)  ATTRIBUT_DE(monnaie, état)
NMOD(état, membre)                PREPOBJ(1er janvier 1999, à compter de)
DEFINITION(euro, monnaie)         DATE(1er janvier 1999)
...
```

and the slot 'ANSWER' is instantiated by *euro*.

2.2.2 Complex questions

Complex questions ('how', 'why', etc.) do not expect any short answer. On these kinds of questions, the system behaves more as a passage retrieval system. The paragraphs containing the more syntactic dependencies in common with the question are selected. Among them, the best-ranked is the one that is returned first by Lucene. For example:

0155 - *Pourquoi convient-il de revoir l'architecture du réseau Animo ?*
(*Why should the structure of an ANIMO network be revised?*)

- Syntactic dependencies and NE tagging:
  `VMOD(convenir, revoir)` `DEEPOBJ(revoir, architecture)`
  `ATTRIBUT_DE(architecture, réseau)` `NMOD(réseau, animo)`
- Question type: **complex (why)**
- Expected type: $\emptyset$

The following passage is selected because all the dependencies of the question are found in the passage:

Passage: *considérant que, à la suite de différents travaux effectués dans le cadre communautaire, notamment lors d'études et de séminaires, il convient de revoir l'architecture du réseau Animo afin de procéder à la mise en place d'un système vétérinaire intégrant les différentes applications informatisées ;*
(*Whereas, as a result of the work carried out at Community level in the course of studies and seminars, the structure of the ANIMO network should be revised so that a veterinary system integrating the various computer applications can be introduced;*)

```
DEEPSUBJ(convenir, il)          VMOD(convenir, revoir)
DEEPOBJ(revoir, architecture)   ATTRIBUT_DE(architecture, réseau)
NMOD(réseau, animo)
PREPOBJ(procéder, afin de)      VMOD(procéder, mise)
PREPOBJ(mise, à)                NMOD(mise, place)
...
```

2.3 Scoring

FIDJI's scores are not composed of a single value, but of a list of different values and flags. The criteria are listed below, and are presented in decreasing order of importance:

- As we said, a paragraph containing an extracted short answer will be prefered if it exists.
- Named entity value (appropriate NE value or not – only for factoid questions).
- Keyword rate (between 0 and 1, the rate of question major keywords present in the passage: proper names, answer type and numbers).
- Answer type value (appropriate answer type or not – only for factoid questions).
- Frequency weighting (number of extracted occurrences of this answer – only for factoid questions).
- Document ranking (best rank of a document containing the answer, as returned by the search engine. If this case, the lower the better).

3 Results

We present the results Table 1 by types of questions. Only one answer per question was allowed, so the values simply correspond to the rate of correct answers for each question type.

Question type	Number of questions	Correct answer
Factoid	116	36.2 %
Definition	101	15.8 %
List	37	16.2 %
"How"	76	22.4 %
"Why"	170	40 %
TOTAL	500	30.4 %

Table 1: FIDJI results by question types.

Results are lower than former campaigns' scores, especially concerning factoid and definition questions.

Looking carrefully at the results shows that, in these particular documents, using syntactic dependencies as the main clue to choose paragraph candidates is not always a good way to find out a relevant passage. This is especially true for complex questions, but not only. Indeed, the selection of the paragraph containing the most question dependencies often leads to the introduction of the document or to a very general paragraph containing poor information.

For example:

0006 - *What is the scope of the council directive on the trading of fodder seeds?*

is answered by

* COUNCIL DIRECTIVE of 14 June 1966 on the marketing of fodder plant seed (66/401/EEC) *

containing many dependencies but answering nothing, while a good result was later in the same document, but with an anaphora:

* This Directive shall apply to fodder plant seed marketed within the Community, irrespective of the use for which the seed as grown is intended. *

Dependency relations are still useful to find the good document, but often fails to point out to the correct paragraph.

Also, JRC-Acquis corpus uses a different register of language than usual corpora such a Web or newspapers. Question as well as document analyses suffered from the specific expressions and structures used by French texts, and especially for definitions. Definitions, quite easy to detect in newspaper corpora, have been poorly recognized for this evaluation.

4 Conclusion

We presented in this article our participation to the campaign resPubliQA 2009 in French. We adapted our syntactic-based QA system FIDJI in order to produce a single long answer in the form of JRC-Acquis tagged paragraphs. Results showed that syntactic analysis should be used in different manners according to the type of tasks and questions. A careful look at our system's errors should enable improvement of robustness of the search by applying contextual strategies.

References

[1] Salah Aït-Mokhtar and Jean-Pierre Chanod. Incremental finite-state parsing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 72–79, Washington, DC, USA, 1997. Morgan Kaufmann Publishers Inc., San Francisco, California, USA.

[2] Erik Hatcher and Otis Gospodnetić. *Lucene in Action*. Manning, 2004.

[3] Véronique Moriceau and Xavier Tannier. Étude de l'apport de la syntaxe dans un système de question-réponse. In *Actes de la Conférence Traitement Automatique des Langues Naturelles (TALN 2009, poster)*, Senlis, France, jun 2009.

[4] Véronique Moriceau, Xavier Tannier, and Brigitte Grau. Utilisation de la syntaxe pour valider les réponses à des questions par plusieurs documents. In *Proceedings of workshop on COnférence en Recherche d'Information et Applications, CORIA*, Presqu'île de Giens, France, 2009.