# Justification of answers by verification of dependency relations - The French AVE task

Véronique Moriceau, Xavier Tannier, Arnaud Grappy, Brigitte Grau
LIR Group - LIMSI (CNRS)
first_name.last_name@limsi.fr

**Abstract**

This paper presents LIMSI results in Answer Validation Exercise (AVE) 2008 for French. We tested two approaches during this campaign: a syntax-based strategy and a machine learning strategy. Results of both approaches are presented and discussed.

### Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing—*Linguistic processing* ; H.3.3 Information Search and Retrieval

### General Terms

Measurement, Performance, Experimentation.

### Keywords

Question answering, Syntactic analysis, Machine learning.

## 1 Introduction

This paper presents LIMSI results in Answer Validation Exercise (AVE) 2008 for French. In this task, systems have to consider triplets (question, answer, supporting text) and decide whether the answer to the question is correct and supported or not according to the given supporting text.

We tested two approaches during this campaign:

- A syntax-based strategy, where the system decides whether the supporting text is a reformulation of the question.
- A machine learning strategy, where several features are combined in order to validate answers: presence of common words in the question and in the text, word distance, etc.

Sections 2 and 3 present respectively both approaches while results and comments concerning our systems and the general task are given in Section 4.

## 2 A syntax-based strategy: FIDJI

Most of question-answering (QA) systems can extract the answer to a factoid question when this one is explicitly present in texts, but in the opposite case, they are not able to combine different pieces of information for producing an answer. FIDJI[1] (Finding In Documents Justifications and

[1]CONIQUE Project ANR-05-BLAN-008501.

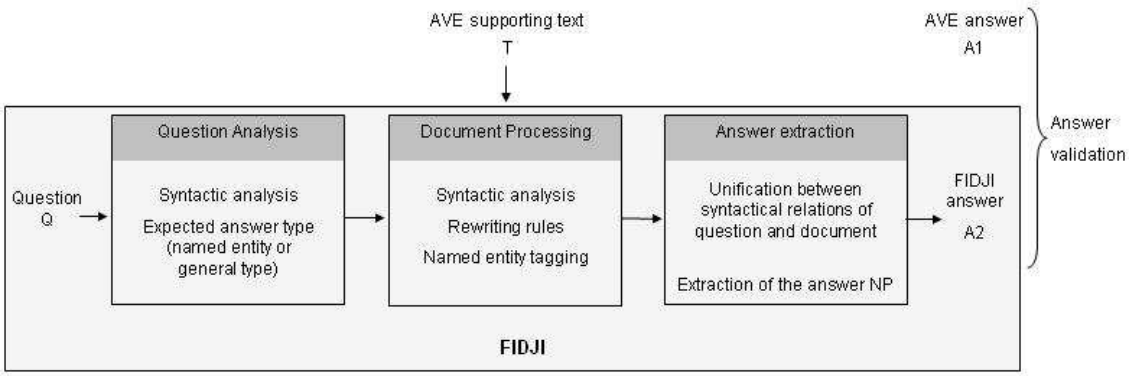


Figure 1: Architecture of FIDJI and answer validation system

Inferences), an open-domain QA system for French, aims at going beyond this insufficiency and focuses on introducing text understanding mechanisms relying on inferences. FIDJI uses syntactic information, especiallly dependency relations: The goal is to match the dependency relations derived from the question and those of the potential answer, as in [8]. Figure 1 presents the architecture of FIDJI and its adjustments for the AVE task. The system is at its beginning and should evolve a lot in the future.

## 2.1 Processing of supporting texts

Our system relies on syntactic analysis provided by Syntex [3], a dependency parser for French. Syntex is used to parse questions as well as the document collection from which answers are extracted. Syntex outputs are here given in an easily readable format. The named entities of documents are also tagged. For the AVE task, supporting texts are considered as documents from which answers have to be extracted.

### 2.1.1 Syntactic analysis

To apply our system to the AVE competition, all supporting texts are syntactically parsed. The approach is to detect, for a given question (Q)/answer ($A_{ave}$)/supporting text (T) tuple, if all the characteristics of the question Q can be retrieved in the text T. Then, the answer proposed by our system ($A_{fidji}$) is compared to $A_{ave}$: if $A_{ave}$=$A_{fidji}$, the answer is validated and justified by T. To determine if the characteristics of the question Q can be retrieved in text T, FIDJI detects syntactic implications between Q and T. There are mainly two cases:

1. There is an exact matching between syntactic dependencies of Q and T: the NP which unifies with the variable of the question representing the answer is extracted:

   Example:

   Q141: `Qui est Lionel Mathis ?` *(Who is Lionel Mathis?)*
   `attribut(ANSWER, Mathis)`
   `NNPR(Mathis, Lionel)` *(proper noun relation)*

   Text: `Lionel Mathis est un footballeur français né le 4 octobre 1981 à Montreuil-sous-Bois (France)` *(Lionel Mathis is a French footballer born...)*
   `attribut(footballeur, français)`
   `attribut(footballeur, né)`
   `attribut(footballeur, Mathis)`
   `NNPR(Mathis, Lionel)`

```
SUJ(Verbe, NP1)        SUJ(Verbe, NP2)
OBJ(Verbe, NP2)   ⇒    AUX(être, Verbe)
                       modif_par(Verbe, NP1)
```

Figure 2: Example of rewriting rule: active to passive voice.

...

The lemma which unifies with the variable ANSWER of the question is "footballeur" (*football player*) and the extracted NP is "footballeur français" (*French football player*). The NP is composed of the head and its basic modifiers (noun complements and adjectives).

2. There are syntactic implications between Q and T. Because of syntactic variations, information in texts are not always expressed in the same way as in questions. Thus, reasoning over syntactic dependency relations is essential. As in [2], we have implemented about 30 rewriting rules to account for passive/active voice, nominalization of verbs [7], appositions, coordinations, etc.

   Rewriting rules are applied to parsed supporting texts. In this way, whatever the syntactic form of the question, the system is likely to find an equivalent syntactic formulation in the given supporting text.

   Example:

   Q105: `Quelle ville a été secouée par un tremblement de terre le 17 janvier ?` *(Which city was hit by an earthquake on the 17th of January?)*

   ```
   DATE( , 17 janvier)
   SUJ(secouer, ANSWER)
   AUX(être, secouer)
   modif_par(secouer, tremblement)
   attribut_de(tremblement, terre)
   ```

   Text: `Le tremblement de terre qui a secoué, lundi 17 janvier à 13 h 31, le nord de la région de Los Angeles ne serait pas associé directement à la fameuse faille de San-Andreas qui balafre la Californie sur des centaines de kilomètres.`

   ```
   SUJ(secouer, tremblement)      SUJ(secouer, nord)
   OBJ(secouer, nord)        ⇒    AUX(être, secouer)
                                  modif_par(secouer, tremblement)
                        DATE( , 17 janvier)
                  attribut_de(tremblement, terre)
                     attribut_de(nord, région)
                  attribut_de(région, los angeles)
   ```

   The left column gives the dependency relations of the supporting text which have also been rewritten into passive voice (right column). In this example, all relations of the question match with the relations of the supporting text.

### 2.1.2 Named Entities

The named entities of texts are tagged with about 20 named entity types (person, organization, location, nationality, date, number, etc.) [4]. This tagging, combined with the question analysis, is useful to check the matching between the named entity type expected by the question and the extracted answer type. For example, the following question expects an answer of type LOCATION:

Q113: `Où Barbara Hendricks a-t-elle donné son premier concert de l'année ?` *(Where did Barbara Hendricks give her first concert of the year?)*

Text: `<enamex type="PERSON">Barbara Hendricks</enamex> a donné son premier concert de l'Année nouvelle à <enamex type="LOCATION-CITY">Sarajevo</enamex>.`

All the syntactic dependency relations of the supporting text match with those of the question and the expected answer type matches with the type of the extracted answer "Sarajevo".

## 2.2 Answer extraction with FIDJI

The answer extraction is based on sentence-level analysis. Sentences having the maximum number of dependencies in common with the question (in other words: The minimum number of missing relations) are considered. For each sentence:

- If the slot for the answer in question dependencies is unified in the candidate sentence, then the corresponding word is extracted (see section 2.1.1).
- If not, named entities having the expected type (if existing) are selected in the sentence and the sentence before.

Weights are attributed in order to rank answers; As they are not used in AVE, they are not described here.

## 2.3 Answer validation for AVE: heuristics

At the current state of our system, a few heuristics are used to validate an answer. The different modules described above provide information concerning:

- In some cases, the matching (or not) between, on the one hand, the expected named entity type and answer type[2] and, on the other hand, $A_{ave}$;
- The rate of syntactic dependencies from the question that are also found (after rewriting) in the passage.

The answer type checking is efficient when seeked on a large collection of documents. It is quite rare to be able to confirm it in a single passage. For this reason, we did not use this clue at all in our AVE run[3]. Finally, an answer was validated only if:

1. It was also an answer suggested by FIDJI,
2. The NE type was the proper one,
3. The rate of missing dependencies was under a given threshold. This threshold has been experimentally set to 30% by testing different configurations on AVE 2006 and AVE 2007 collections.

These heuristics have been chosen in order to maximize precision. For an exercise such as AVE, we think that precision is more important than recall. The second run, presented in Section 3, has been designed in order to improve recall rate.

**Validated *vs.* selected.** If only one answer was approved by the system, it was marked as SELECTED. When more than one answer were validated, the best one (i.e. the one returned at the best position by FIDJI) was marked as SELECTED and the others as VALIDATED.

[2] An explicit type suggested by the question, as "prime minister" in *"Which prime minister has..."*. Again, we do not enter into details for this part because we do not use it in AVE.

[3] But our second run presented in next section checks the answer type in a very different way, through Wikipedia pages.

**Wikipedia.** Wikipedia passages were identified by their titles. It is a well-known observation that Wikipedia articles concerning persons contain very long-distance pronominal anaphoras ; for most of them, these references can be resolved by replacing the pronoun by the article title. We used this simple trick with pronouns "il" (he) and "elle" (she).

Results are presented and discussed in section 4.

# 3 FRASQUES as an entry of a machine learning system

The second system follows a machine learning approach and applies the question-answering system FRASQUES [6] in order to compute some of the learning features. The learning set is extracted from the data provided by AVE 2006 and contains 75% of the total data.

The chosen classifier is a combination of decision trees with the bagging method. It is provided by the WEKA[4] program that allows to test a lot of classifiers.

The next sections present the different features.

The specific features based on the vocabulary are presented in [1], while [5] shows and evaluates these features and presents the machine learning method.

## 3.1 Common terms

If a passage contains many terms of the question then it ought to be about the same topic and would probably contain the answer. Thus, the first feature is the rate of terms of the question that are in the supporting text, with or without lexical variations. These variations are recognized by FASTR [7].

Among question terms, some play a more important role and are supposed to be found in the supporting passages or have to be verified. Four particular roles are distinguished in the questions:

- **Focus:** The focus is the entity about which the question is asked and either a characteristic or a definition of this entity has to be searched. In *"Which is the political party of Lionel Jospin?"*, the focus is "Lionel Jospin".
- **Answer type:** When the specific answer type is explicit in a question and recognized in a passage, it allows the system to check that the proposed answer fits the expected type by applying some syntactic rules. In the previous question, the expected type is "political party".
- **Main verb:** The verb in the question that has an important role because it corresponds to an action or a fact.
- **Bi-terms:** A bi-term is made of two words syntactically linked as "Nobel Prize". If a bi-term is in the question and in the passage, then the words are likely to have the same meaning.

Each of these terms constitutes a feature given to describe a passage. These elements are automatically recognized by the question analysis module of FRASQUES.

## 3.2 Answer verification

Another feature is based on the answer extracted from the passage by FRASQUES. If the FRASQUES answer is equal to the answer to judge, the latter is probably correct.

The extraction strategy of FRASQUES depends on the expected type of answer. If this type is a named entity, the entity of the expected type which is closest to the question words is selected. Otherwise, patterns of extraction are used. These patterns express the possible position of the answer with respect to the question characteristics such as the focus or the expected type of the answer.

[4] WEKA : http://www.cs.waikato.ac.nz/ml/weka

### 3.3 Longuest common chain of words

This feature relies on the proximity of the common terms. The system looks for the longuest common string of consecutive words in the passage and the hypothesis without considering their order. The hypothesis corresponds to the affirmative form of the question concatenated with the answer.

To compute this chain, the strategy is the following:

1. In order to facilitate the comparison between the text and the hypothesis, the words are normalized (lemmatization and bringing of synonyms together).
2. The algorithm looks for the longest groups of adjacent words common to the question and the hypothesis.
3. A string is initialized with each of these groups. Each string will grow by concatenating adjacent groups (or groups that are separate by allowed items like a comma or a determinant). The groups can also be separated by one plain word counting as a bonus. Only one bonus is allowed.

   For example, when comparing the strings "Elisabeth 2, l'actuelle reine" (*Elisabeth II, the current queen*) and "Elisabeth 2 reine" (*Elisabeth II queen*), "Elisabeth 2" and "reine" are joigned because they are separated by a determinant ("l' "), a comma and only one other word ("actuelle").
4. The longest chain is selected and the value of the feature is the ratio between the number of words in the chain and the number of words in the hypothesis.

### 3.4 Checking the answer type with Wikipedia

A lot of questions expect an answer of a specified type. For example the question *"What sport did Zinédine Zidane practice?"* expects a kind of sport as answer. To verify the type of the answer, we use the encyclopaedia Wikipedia[5].

The hypothesis is that if the type can be found in the Wikipedia page corresponding to the answer, we consider that the answer corresponds to the expected type.

The method looks for the type in the Wikipedia page whose title contains the answer. If the type is found, the value of the feature is 1 else it is 0. For questions without expected type, the value is -1.

### 3.5 FIDJI features

Some features coming from FIDJI are also added:

- Whether FIDJI validated, ignored (known by below the threshold) or rejected (unknown) the answer;
- Good or bad named entity type;
- Rate of missing dependancies in the passage.

## 4 Results and comments

Official results for our two runs are given in Tables 1.a (run based on syntactic relations) and 1.b (combining different characteristics by the use of a classifier and including as supplementary feature the rate of presence of dependency relations given by FIDJI).

[5] Wikipedia : http://fr.wikipedia.org

| a. Results for FIDJI alone run | |
|---|---|
| F measure | 0.57 |
| Precision over YES pairs | 0.88 |
| Recall over YES pairs | 0.42 |
| qa accuracy | 0.19 |
| estimated_qa_performance | 0.29 |

| b. Results for ML run with FIDJI | |
|---|---|
| F measure | 0.61 |
| Precision over YES pairs | 0.75 |
| Recall over YES pairs | 0.52 |
| qa accuracy | 0.23 |
| estimated_qa_performance | 0.32 |

Table 1: Official AVE 2008 results.

| | |
|---|---|
| F measure | 0.63 |
| Precision over YES pairs | 0.67 |
| Recall over YES pairs | 0.60 |

Table 2: Results without dependency relations

Table 2 shows the results obtained when we do not include characteristics coming from the FIDJI system. Finally, a baseline for French AVE, provided by the organizers and presented in Table 3, corresponds to the strategy consisting in answering YES to each pair.

Now, the question is to know the signification of this test set. This year, the French test set is made of 199 triples, built from 108 different questions. There are 1.8 triples in average for each question : 39 questions have 1 answer to justify, 47 questions have 2 answers, and 22 questions have 3 proposed answers. The ratio of validated pairs is 29%, that is to say that only 52 triples are correct. These are the results provided by one participant to the monolingual French QA task and the bilingual tasks with French as target. Thus, the answers result from a single system.

If we compare this test set to the test set provided for French in 2006, there were 5 different systems that have given 3200 answers to 190 questions : among them, 627 answers were justified.

So, the current test set could only measure the ability of a AVE system to evaluate the results of one QA system, which is the best system in this language, but cannot allow to measure its ability in a general exercice of answer validation. Moreover, are the results really significant when they are calculated over a total of 50? One answer is equivalent to 2 points.

The goal of an evaluation campaign is generally twofold : to provide ressources allowing to develop systems able to solve a task and comparing the different approaches developed for this task. In order to tend towards these goals, the French AVE test set could not only be made of the results of the current QA tracks. It ought to be completed so that the number of examples will be significant and the phenomena to treat representative of a task, and not of a system.

Another important point concerns the definition of what a justification of an answer to a question is. Which pieces of information must the passage contain? In AVE, it seems that if the correct answer is in the passage, it is validated, even if the topic is only present with an anaphora, as in the following pair:

Q: Combien la ville de Colombo comptait-elle d'habitants en 2001? (*How many inhabitants are there in Colombo in 2001 ?*)
A: 377 396
J: La ville compte 377 396 habitants en 2001 pour 2 234 289 dans l'agglomération et c'est la ville la plus peuplée du Sri Lanka, ainsi que le c¡ur de l'activité commerciale de ce pays. (*The town has 377 396 inhabitants in 2001 ...*)

| | |
|---|---|
| F measure | 0.45 |
| Precision over YES pairs | 0.29 |
| Recall over YES pairs | 1 |

Table 3: baselines Results

Whithout reading the document that contains this passage, it is not possible to assert that "La ville (*the town*)" is "Colombo", even if the name of the document is COLOMBO. The only name of a Wikipedia page cannot allow to verify the reference of the anaphora.

# 5 Conclusion

We have presented in this paper two strategies for deciding if an answer to a question is justified by a given extract of text.

The first is based on a syntactic approach in order to verify that not only the vocabulary is similar between the question and the passage, but also that this vocabulary is used in the same meaning. This is done by verifying the similarity of the relations between the corresponding terms. Some heuristics are then chosen for deciding if a passage justifies or not an answer.

Such an approach has good performances at the precision level, but the recall remains low, because of errors done by the syntactic parser, as in all these kind of approaches. So, we also tested another approach consisting in deciding if a passage is a justification or not according to a set of features. The decision is the result of a classifier, automatically trained. These last approach has been developed last year, and we have added this year a new feature based on the dependency relations. We have to test our results on other corpora in order to validate the gain that this feature seems to bring out.

# References


[1] A.Ligozat, B. Grau, A. Vilnat, I. Robba, and A. Grappy. Lexical validation of answers in question answering. In *Proceedings of workshop on Web Intelligence, WI*, Silicon Valley, 2007.

[2] G. Bouma, I. Fahmi, J. Mur, G. van Noord, L. van der Plas, and J. Tiedemann. Linguistic knowledge and question answering. *Traitement automatique des langues*, 46, 2007.

[3] D. Bourigault and C. Fabre. Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaire*, 25, 2000.

[4] F. Elkateb. Extraction d'entités nommées pour la recherche d'informations précise. In *4éme congrès ISKO-France, L'organisation des connaissances*, Grenoble, 2003.

[5] A. Grappy, A. Ligozat, and B. Grau. Evaluation de la réponse d'un système de question-réponse et de sa justification. In *Proceedings of workshop on COnférence en Recherche d'Information et Applications, CORIA*, Trégastel, 2008.

[6] B. Grau, A. Ligozat, I. Robba, A. Vilnat, and L. Monceaux. Frasques: A question-answering system in the equer evaluation campaign. In *Proceedings of workshop on Language Ressource Evaluation Conference, LREC*, Genoa, 2006.

[7] C. Jacquemin. A symbolic and surgical acquisition of terms through variation. In *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Heidelberg, 1996.

[8] B. Katz and J. Lin. Selectively using relations to improve precision in question answering. In *Proceedings of workshop on Natural Language Processing for Question Answering, EACL*, Budapest, 1999.