Fine-grained linguistic evaluation of question answering systems

Sarra El Ayari *, Brigitte Grau * **, Anne-Laure Ligozat * **

LIMSI - CNRS *
ENSIIE **
firstname.lastname@limsi.fr

Abstract

Question answering systems are complex systems using natural language processing. Some evaluation campaigns are organized to evaluate such systems in order to propose a classification of systems based on final results (number of correct answers). Nevertheless, teams need to evaluate more precisely the results obtained by their systems if they want to do a diagnostic evaluation. There are no tools or methods to do these evaluations systematically. We present REVISE, a tool for glass box evaluation based on diagnostic of question answering system results.

1. Introduction

Many campaigns were organized to evaluate natural language processing systems: TREC (1998) for question answering systems, MUC (from 1987 until 1996) for information extraction, EASY (2004) for syntactic parsers, etc. The outcomes of these campaigns are twofold:

- about corpora: systems can be assessed and improved;
- about evaluation of systems i.e. overall evaluation of their approaches: each campaign should ideally prove the relevance and performance of given strategies (Gillard et al., 2006).

In question answering, these results cannot always be exploited at their best. Corpora, built by the organizers and QA participants, consist of:

- questions (200 to 500): factual, definition, complex, boolean;
- precise answers: short string, correct or not;
- possibly, the supporting texts given by participating QA systems: documents or snippets from which the answer was extracted that allow to recognize that the answer is correct.

Yet, using these corpora to improve a system may not be useful. First, the documents are plain text: no annotation is given, for example annotation of the answer string, or linguistic variations of the question. Moreover, they may not be representative of the difficulty to answer questions: some questions are not answered by systems and documents that contain the answer with the same phrasing as the question are better recognized by systems and are then the only returned documents.
In order to exploit more acutely these corpora, they should thus be annotated with relevant information, such as the answer string, question terms or variants and the different characteristics extracted from questions that have to be retrieved in answer passages.
Moreover, computing a system performance cannot be the only way to evaluate a QA system because this type of evaluation is global and cannot provide a real understanding of the system performances relative to some specific linguistic phenomena. Research teams still need diagnostic evaluations to know the reasons of their successes and their failures that are related to the capacities of the systems to handle linguistic properties and to elaborate resolution strategies. However there are no tools or methods to produce systematic evaluations of linguistic criteria for such systems.
Thus, we envisage another method of system evaluation: glass box evaluation, which consists in evaluating the results produced by each component of a system in order to measure their relevance compared to the whole process. Some studies present different kinds of glass box evaluations applied to question answering systems: by modification of the system architecture (Costa and Sarmento, 2006), (Moriceau and Tannier, 2009) or by controlling the flow of data (Nyberg et al., 2003), (Kursten et al., 2008). Nevertheless the organisation into sequences of the different components makes the linguistic criteria difficult to track because they are not limited to one component: they appear on the complete process.
The paper will present a state of the art section 2.. Then we will focus on linguistic properties that QA systems have to handle section 3. and how they can be studied. Finally, section 4. presents our framework REVISE.

2. State of the Art

Glass box evaluation allows researchers to measure the contribution of each component of a modular system to the final results. As a consequence, a glass box evaluation enriches a black box evaluation (Sparck Jones, 2001): the method to choose depends on what needs to be evaluated.
Nevertheless, even if this kind of evaluation is needed to improve systems, there are few papers about glass box evaluations. However, two kinds of components evaluation can be distinguished:

2.1. Modifying the system architecture

The first evaluation method for evaluating components consists in disconnecting a component and measuring the final results obtained. This approach was used by (Costa and Sarmento, 2006) on the ESFINGE system, which only works on the Portuguese language. Their point of view is

that evaluating the achievement of each component is essential for measuring their impact on the final results and that they need to identify which component is needed and which is obsolete.
Tomas et al. (Tomas et al., 2005) propose a tool based on XML technology for making the integration, the combining and the evaluation of components built on different ways easier. They tried to facilitate the development of their system by:

- replacing components without modifying the system itself (the process is listed in an XML file);
- testing a component regardless of the whole process.

2.2. Controlling the execution

The second type of approach for evaluation is exemplified by the JAVELIN system (*Justification-based Answer Valuation through Language Interpretation*) (Nyberg et al., 2003). It is a question answering system in which a component can control the executing process and the data used. Different strategies of resolving questions can be tested and then added to the system. They developed a planner component which can automatically select different versions of the components used in order to find the best one.
Nevertheless, the question of a generic tool for glass box evaluation on question answering systems is not fulfilled, and as the granularity degree of evaluation is the component level, these approaches do not allow a fine grained evaluation of the resolution of some linguistic phenomena.

3. Evaluating linguistic phenomena

A question answering system allows a user to ask a question in natural language (not with keywords) and provides a precise answer extracted from a text. For example the system must answer *four* to the question *How many people were the Beatles?*
In order to extract an answer, question answering systems have to connect the question and the answer formulations in a passage and have to deal with linguistic variations for that.

3.1. Examples of linguistic variations

We present here some phenomena of linguistic variations in order to measure their involvement for the process of automatically answering a question.
We show different sentences containing a precise answer[1] and a variation form[2] of the verbal group for the question *Who did Michael Jackson marry in March 1994?*

1. Michael Jackson and **his wife**, [Lisa-Marie Presley], arrived Wednesday in Budapest.
2. The American star Michael Jackson **got married** with [Elvis Presley's daughter] on March.
3. The **wedding** of Michael Jackson and [Lisa Marie Presley Keough] take place on March 1994.
4. Yes, [Lisa Presley] confirms that she **has tied the knot** with Michael Jackson.

We can identify a nominal form *his wife*, a compound form *got married*, a nominalization *wedding* and a locution *has tied the knot* of the verb *marry*.
These cases of variations are common and complicate the recognition of the question information rephrasing. Thus, our aim is to provide a solution that allows the evaluation of the resolution of such problems by use of NLP techniques.

3.2. Needs for precise evaluation

Evaluation of NLP techniques only with a black box evaluation is not helpful enough to improve systems, since it does not evaluate the specific processing of these fine-grained NL properties (Popescu-Belis, 2007). Analysis of examples such as the preceding ones shed light on important steps and analyses necessary to constitute a glass box methodology of evaluation. These evaluation needs are:

1. Corpus analysis
   The observation of the data produced by a question answering system as a whole corpus requires a complete access to them. It has to be possible to annotate this data, categorize it according to the phenomena observed and visualise it. It is also necessary to take into account the qualitative aspect of language and we have to be equipped for this.

   Depending on these elements, we need tools for:

   - selection of the data to be analysed with fine-grained criteria;
   - observation of linguistic properties;
   - tagging of data;
   - modification of data.

   (Cohen et al., 2004) has similar goals and propose a tool for defining corpora for studying biological Named Entities (NE) that are representative of chosen linguistic features. However, this tool is designed only for evaluating NE recognizers.

2. Performance evaluation
   We cannot evaluate a system without taking into account the consequences of modifications on the final results. To ensure that fine-grained modifications allow a better processing, final results must be evaluated. For example, the modification of a rule to handle semantic variation of a term can generate noise and lower the number of correct precise answers extracted (Berthelin et al., 2001). Thus, it is essential to check that the changes do not alter final results (due to the sequential processes).

   For this, we need tools for:

   - process launching during the modification process;
   - result evaluation and comparison of two successive runs.

[1] The precise answer is noticed into brackets.
[2] The variation appears in bold.

We have integrated all these functionalities in a tool REVISE that allows carrying out a diagnostic study of a complex NLP system. REVISE is dedicated to evaluating question answering systems by storing intermediary results, visualising them, annotating the errors and exporting data to launch again the system with the modified results. It also allows creation of sub-corpora depending on linguistic phenomena.

4. REVISE: a glass box evaluation framework

REVISE, acronym for *Research*, *Extraction*, *VISualization* and *Evaluation*, is a tool for evaluating the intermediary results produced by any question answering system (El Ayari, 2009)[3], (El Ayari and Grau, 2009). The approach consists in searching criteria extracted by the question analysis module in the sentences selected by the system that contain the precise answer. First we will present a general overview of a QA system, in order to show that our approach is generic and can be applied to many systems of QA, then we will present the architecture of REVISE and detail its use.

4.1. Architecture of QA systems

QA systems generally follow a pipeline architecture in order to realize the following treatments: question analysis, document search and tagging, passage selection and answer extraction. Our system FRASQUES (Grau et al., 2006) obeys to this kind of architecture. In order to show that the evaluation framework we propose can be applied to these kinds of systems, we will describe the role of each component and what can be done for their evaluation.

4.1.1. Question analysis

Question analysis extracts information about the question, which are given to the other components. If the criteria are incorrect, the possibility of extracting a good answer is reduced. All systems extract the expected answer type (a named entity type or a type of concept), and question terms. These two kinds of criteria constitute the basis for selecting relevant texts or passages, in which terms and named entities are recognized. Additional linguistic properties can be handled to develop more precise matching processes between question and passages as syntactic analysis, that can be partial or not, deep analysis or shallow analysis.

The idea is that if all the properties found in the question are retrieved in a passage, then it is supposed to contain a reformulation of the information given in the question and give the answer to this question.

FRASQUES extracts the category of the question (definition, instance...), the semantic type (answers hyperonym) and the focus (entity on which the question is asked).

4.1.2. Selection of passages

The second module searches documents or passages where the terms of the question, and possibly their linguistics variations, are present. These texts are processed a minima by a Named Entity Recognizer (NER). According to the size of selected text, an additional step, as in FRASQUES, can be added that consists in selecting sentences which may contain the answer. The sentences are weighted according to their similarity with the question, similarity computed by taking into account the presence or not of the criteria extracted from the analysis of the question.

4.1.3. Answer extraction

The last step consists in extracting the precise answer from the sentences by applying extraction patterns or syntactic matching or selecting the expected named entity, or applying all these criteria in combination. FRASQUES is based on extraction patterns that are determined with respect to the question category and are based on pivot terms: focus, main verb or semantic type.

As a result, evaluating the accuracy of each component needs an access to the intermediate results to estimate their contribution. In this way, it will be possible to follow the treatment of specific phenomena all along the process by storing all the question characteristics resulting from the question analysis module and retrieving them in the results of the analysis of selected passages.

4.2. Architecture of REVISE

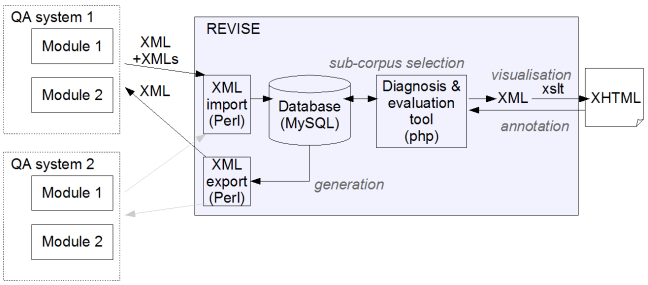

Figure 1: REVISE architecture

Figure 1 shows the architecture of REVISE. A relational database stores the data produced by a question answering system. The relational schema of the database depends on the structuring of this data, thus of its XML structure.

Database. In order to use REVISE, QA systems have to define an XML schema that explicits that structure. The compulsory part of the XML structure lies in the necessity of describing the results of two processes: an XML substructure to represent the question analysis results and another to represent the passage analysis results. Different tables thus correspond to each level of analysis (question analysis, passage analysis, words, lemmas and their synonyms, precise answers). Figure 2 shows the relational schema for the FRASQUES intermediary results. The notion of run in the tables allows to store different versions of the results (another evaluation campaign, another system version) and to compare them.

Table schemas depend on each system and their attributes are the characteristics pro vided by the different modules: scores of selected sentences, features of the questions, etc. The intermediary results are exported in XML format from QA systems and imported into the database. The XML

[3]http://www.limsi.fr/Individu/sarra/these

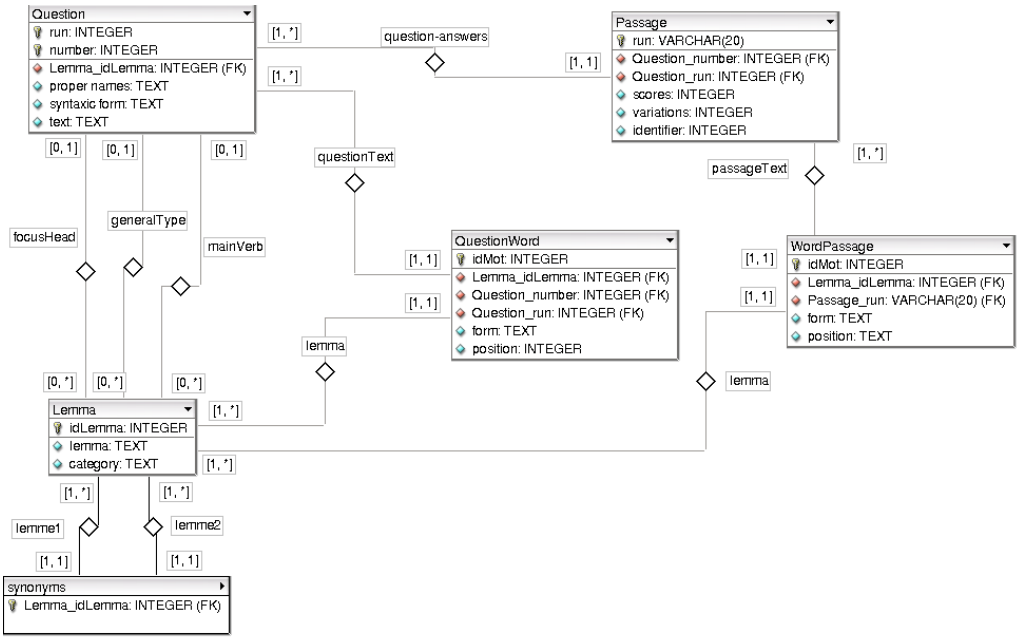

Figure 2: Excerpt of the database schema for FRASQUES results

schemas (XMLs) associated to these XML files allow the definition of the corresponding tables in the database.
At the moment, some queries can be predefined, according to the database schema or can be defined interactively in SQL by each user and stored for reuse. In a second step, we will develop an interface to guide the user when conceiving these queries, which will avoid her to know SQL.
Visualization. This database allows a user to visualize or count phenomena by defining queries that can be restricted to all the linguistic criteria that are explicit in the intermediary results. The tool can also generate an output of the SQL queries in XML format. In this way, it ensures interoperability with other tools.
The visualization of data (fig. 3) has an important part in evaluating NLP approaches as it allows to have an overall view of the questions and passages, or if a phenomenon has been handled correctly. We bring to the foreground linguistic information by highlighting it according to contextual criteria either predefined or given by the user. For example, REVISE can colour words according to their POS tagging or colour the synonyms of some types of elements extracted from the questions. The vizualisation process makes use of XSLT style sheets.
Annotation. Lastly, REVISE allows a user to modify or annotate the data stored into the database in order:

- to test the execution of a component by correcting the results of the previous one and run the system with modified results without having to modify the component itself. It allows a simulation of the effects entailed by modifying some linguistic processes in order to know if a strategy will be adequate or not before implementing it;

- to constitute sub-corpora based on new annotations that become new attributes in the database and can thus be chosen as restriction criteria.

4.3. Functionalities

Related to the needs arising from corpus analyzing and performance evaluation, REVISE allows at the same time two types of work:

1. Diagnosis:

   - Automatic selection of linguistic features;
     Selection by means of the table schemas of the database allows to choose fine-grained type of data: question category, types of words, type of answers, etc.

   - Sub-corpus selection;
     Users can select which data to visualize by choosing precise features: sentences containing correct answers, questions of one category, etc. They can also select sub-corpus according to added annotations. In this way, a user thus can focus on a precise problem.

   - Manual annotation and modification of errors;
     This functionality is needed to identify the errors due to extraction criteria, tagging, syntactic analysis, etc. It allows a quantitative diagnosis by types of error. The given possibility of correcting errors permits to evaluate further components independently of these errors.

   - Evaluation of the frequency of some linguistic phenomena;

   - Evaluation of performance by relating selected passages with the exact answer.
     As we said before, evaluation of results is important to make a reliable diagnosis of system performances.

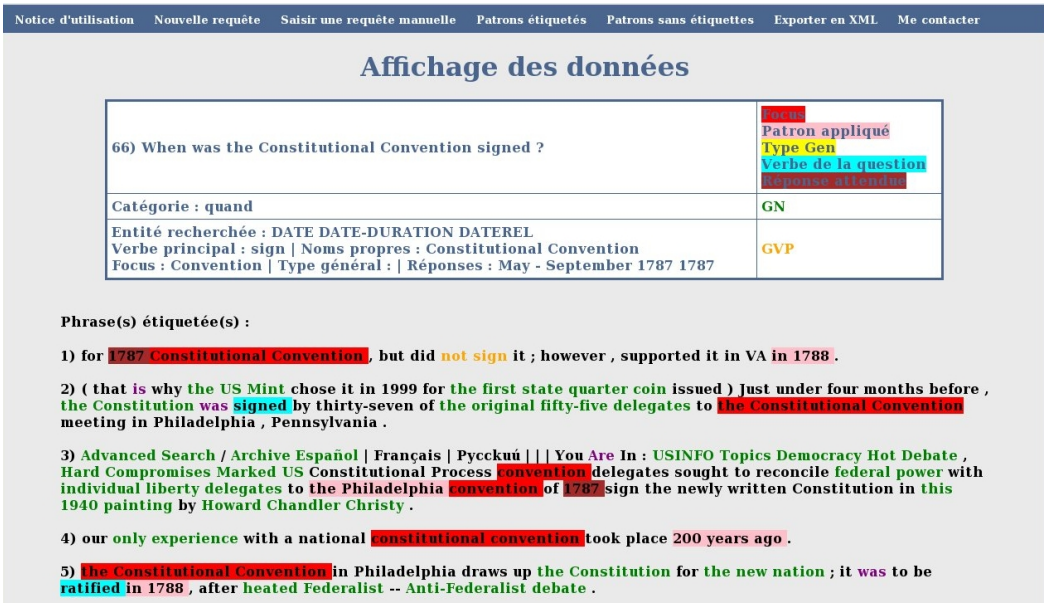

Figure 3: Visualization in REVISE

Diagnosis is essential for an error analysis of a system and for the creation of packs of questions which are related to the linguistic phenomena annotated.

2. Improvement:

   - Simulation with corrected data;
     The modified results are stored into the database and the user can evaluate the relevance of the corrections on the all process. It allows to know which improvement has to be done.
   - Output of modified data in XML format;
     This output is used to re-launch the system.
   - Evaluation of the new results compared to previous one.
     New runs are stored into the database, which allows all the treatment discussed before.

   The functionalities related to the improvement task make the test of criteria easier by simulating a test with modified data before undertaking their automatic processing. It also allows to know which results a system can reach.

5. Examples of uses

REVISE has been used for different purposes on the question answering system FRASQUES and its version for English, QALC (Ferret et al., 2001). The first need was the refining of the focus criterion used in the question analysis component. REVISE allowed us to observe the data, label it and evaluate the results we obtained. The second need was the improvement of answers extraction rules in order to enhance the whole process of answering questions.

5.1. Focus criterion

According to our definition, the focus is an entity which represents either the event or the entity about which the question is asked. For example, the question *What year was Martin Luther King murdered?* has **to murder** as a focus whereas the question *Who is Martin Luther King?* has **Martin Luther King** as a focus. This focus term is interesting because it is the one about which the need of information is required. This information is important for weighting sentences and locating the answer inside them: if a sentence contains this word, the answer will be related to this element.

In the current version of FRASQUES and QALC, the focus chosen by the question analyser is always a noun phrase, as verbs are terms which are highly variable. Thus we made the choice to rather find the subject or the object of the main verb in candidate sentences, although the verb was used for extracting the answer and verifying the existence of relations between triplets (focus, main verb, answer) since it was recognized in the extraction step.

As this approximation leads to discard correct answers in case of the focus absence in the answering sentence (the focus is in the narrow context or referenced by an anaphora), we decided to come back to the original definition, and to explicit which would be the problems to solve, to quantify them and to evaluate the impact of this choice on the current version of the system.

The process for studying the focus relies upon a methodology, which first consists in selecting the questions and labelling them with the type of focus found: event or entity. After that, it is possible to study the sentences selected by the question answering system to verify the existence of a relation between the focus and the answer. In a first approximation, this property was quantified by counting the number of words between the answer and the focus. More precisely, 36 questions were retrieved that were marked as event for focus.

Thus we manually corrected the focus on the results of the question analysis into the database of REVISE according to the two types of focus; then we selected sentences on the

following criterion: possessing both a correct answer and the focus corrected manually (which returned 587 answering sentences). REVISE computed then a mean distance between the focus and the correct answer in the sentences extracted by our question answering system.
This study shows that the two distinct types of focus are more relevant than the single use of entities as focus: average 4 words between focus and answer compared with 8 words with the older definition, and consequently allows us to modify our QA system knowing that it is a good way of improving the whole process.
The first study relied on the presence of the focus in answering sentences with the same lemma. In a second step, we observed the variability of the verb as focus, in order to evaluate the linguistic phenomena to handle. A form was created to allow the addition of a new annotation about the kind of variation found, that was represented by the following types:

- identical to the question verb;
- nominalization (*meeting* for *to meet*);
- synonymy (*wedding* for *marriage*);
- locution (*take place* for *situate*);
- preposition (*of* for indicating *to possess*);

We also added to the sentence representation in the database the variant itself in order to coloured it along with the answer to facilitate the study of the sentences. This lead to the repartition given Table 1.

Phenomena	Presence rate
Nominalization	13%
Synonymy	14%
Locutions	3%
Preposition	2%

Table 1: Variations of the verb as focus

The remaining 77% are cases of focus identical to the question form. This study allows us to evaluate the expected gain when processing each phenomena.

5.2. Extraction rules

The last level of a question answering system consists in extracting precise answers from the passages already selected by the system. The QALC system has a loss of 50% of correct answers, when applied on a corpus provided by the QUAERO project [4], which is collected from the WEB. This study [5] is part of this project which organized a series of evaluations of Question Answering systems on Web Data in 2008 and 2009.
More precisely we developed a particular visualization for sentences where the words of the question are coloured (as shown in figure 3) as well as the extraction rules applied. This kind of tool makes the evaluation of extraction rules easier by showing when they failed and why. We made a study on 195 questions (the other ones were formulated with copula verbs) and the sentences extracted by QALC on the Quaero corpus. We identified different causes of non-application of the rules due to the question-analysis, gathered in table 4.

Errors	Numbers of questions
Verb not found	20
Proper noun not found	8
False focus	95
False semantic type	38
Incorrect syntaxic analysis	46
False type of entity	4

Figure 4: Types of errors

This information is very useful for the improvement of a QA system and cannot be obtained with a black-box evaluation approach. It is the sum of the two types of evaluation which will allow a real diagnostic of a system.
We used REVISE to create new extraction rules for QALC for HOW and WHY question types, that are new types in the evaluation set of questions (Quintard et al., 2010). Our method is based on the observation of answers in context and, after, on the creation of rules and the observation of their application.

6. Conclusion

As a conclusion, the functionalities of REVISE create an open door on system results that can be scanned on a transversal way not depending on the components. REVISE enables a user to measure the impact of a criterion by modifying it and testing its contribution to another component and at the same time to the whole system process. Corpus analysis by visualizing, counting and tagging linguistic phenomena appears to be an essential function to make a precise diagnostic of a system.
In order to spread this kind of methodology over different tasks organised in modular systems, it will be interesting to test REVISE on systems which have several components and are based on the study and the processing of linguistic characteristics both present in the input data of the system and output passages, as for alignment modules for example in translation systems.

7. References

Berthelin, J.-B., Grau, B., and Hurault-Plantet, M. (2001). Two levels of evaluation in a complex nlp system. *Workshop on Evaluation for Language and Dialogue Systems, ACL*.

Cohen, K. B., Tanabe, L., Kinoshita, S., , and Hunter, L. (2004). A resource for constructing customized test suites for molecular biology entity identification systems. In *Association for Computational Linguistics (ACL)*, pages 1–8.

Costa, L. F. and Sarmento, L. (2006). Component Evaluation in a Question Answering System. *LREC*.

El Ayari, S. (2009). *Evaluation transparente du traitement des éléments de réponse à une question factuelle*. PhD thesis, Université de Paris-Sud 11.

[4] http://www.quaero.org/

[5] This work has been partially financed by OSEO under the Quæro program.

El Ayari, S. and Grau, B. (2009). A framework of evaluation for question-answering systems. *European Conference on Information Retrieval (ECIR)*, pages 744–748.

Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., and Jacquemin, C. (2001). Document selection refinement based on linguistic features for QALC, a Question Answering system. *RANLP*.

Gillard, L., Bellot, P., and El-Beze, M. (2006). Question answering evaluation survey. *Language Resources and Evaluation Conference*.

Grau, B., Ligozat, A.-L., Robba, I., Vilnat, A., and Monceaux, L. (2006). FRASQUES: A Question-Answering System in the EQueR Evaluation Campaign. *Language Resources and Evaluation Conference*.

Kursten, J., Wilhelm, T., and Eibl, M. (2008). Extensible retrieval and evaluation framework: Xtrieval. In *Proceedings of Lernen - Wissen - Adaption (LWA-2008)*.

Moriceau, V. and Tannier, X. (2009). Apport de la syntaxe dans un système de question-réponse : étude du système FIDJI. *TALN*.

Nyberg, E., Mitamura, T., Callan, J., Carbonell, J., Frederking, R., Collins-Thompson, K., Hiyakumoto, L., Huang, Y., Huttenhower, C., Judy, S., Ko, J., Kupse, A., Lita, L., Pedro, V., Svoboda, D., and Van Durme, B. (2003). The JAVELIN Question-Answering System at TREC 2003: A Multi-Strategy Approach with Dynamic Planning. In *Proceedings of the Text Retrieval Conference*.

Popescu-Belis, A. (2007). Le rôle des métriques d'évaluation dans le processus de recherche en TAL. In *T.A.L. : Traitement automatique de la langue*, volume 48, pages 67–91.

Quintard, L., Galibert, O., Laurent, D., Rosset, S., Adda, G., Moriceau, V., Tannier, X., Grau, B., and Vilnat, A. (2010). Question answering on web data: the qa evaluation in quæro. *LREC*.

Sparck Jones, K. (2001). Automatic language and information processing: rethinking evaluation. In *Natural Language Engineering*, pages 1–18.

Tomas, D., Vicedo, J. L., Saiz, M., and Izquierdo, R. (2005). Building an xml framework for question answering. *Cross Language Evaluation Forum*.