Question answering on web data : the QA evaluation in Quaero

Ludovic Quintard, Olivier Galibert, Gilles Adda, Brigitte Grau, D Laurent, Véronique Moriceau, Sophie Rosset, Xavier Tannier, Anne Vilnat

Question Answering on web data: the QA evaluation in Quæro

Ludovic Quintard[1], Olivier Galibert[1], Gilles Adda[3]
Brigitte Grau[3,5], Dominique Laurent[2],
Véronique Moriceau[3,4], Sophie Rosset[3],
Xavier Tannier[3,4], Anne Vilnat[3,4]

[1] LNE, Trappes, France [2] Synapse Développement, Toulouse, France
[3] LIMSI-CNRS, Orsay, France [4] Université Paris-Sud 11, Orsay, France
[5] ENSIIE, Evry, France

Abstract

In the QA and information retrieval domains progress has been assessed *via* evaluation campaigns(*Clef, Ntcir, Equer, Trec*).In these evaluations, the systems handle independent questions and should provide one answer to each question, extracted from textual data, for both open domain and restricted domain. *Quæro* 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 *Quæro*. The Quaero project organized a series of evaluations of Question Answering on Web Data systems in 2008 and 2009. For each language, English and French the full corpus has a size of around 20Gb for 2.5M documents. We describe the task and corpora, and especially the methodologies used in 2008 to construct the test of question and a new one in the 2009 campaign. Six types of questions were addressed, factual, Non-factual(How, Why, What), List, Boolean. A description of the participating systems and the obtained results is provided. We show the difficulty for a question-answering system to work with complex data and questions.

1. Introduction

There are multiple paradigms used to search for information. A very popular one, embodied in search engines such as Google, is named *information retrieval*. In that approach, documents matching a user query are returned. The matching is often based on some keywords extracted from the query, and the underlying assumption is that the documents best matching the query provide a data pool in which the users might find information that suits their needs. The queries can be very specific (*e.g. Who is presiding the French Senate?*), or can be topic-oriented (*e.g. I'd like information about the French Senate*). An evolution of that approach is embodied by so-called *question answering* (QA) systems, which return the most probable answers given a specific question. For instance, to the question *In what year was the American Constitution drafted?*, such a system would try to answer *1787*. In the QA and information retrieval domains progress has been assessed *via* evaluation campaigns (Voorhees and Harman, 2005; Forner et al., 2008; Ayache et al., 2006; Mitamura et al., 2008). In these evaluations, the systems handle independent questions and should provide one answer to each question, extracted from textual data, for both open domain and restricted domain.

Quæro[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 *Quæro*, a yearly evaluation campaign on question-answering on web data is organised. The corpus is composed of documents selected from the web from a log of real user requests. The questions are also created by using the same requests. Handling web data is complex due to its varied nature: forum, blog, Wikipedia, journal, spam, etc. This evaluation was performed in 2008 and in 2009, showing a nice improvement in the systems' performance.

2. The Tasks

The purpose of the Quæro QA task is to answer questions on a *Web* data corpus. The fundamental idea is to try to provide more precise answers while staying with an interface reminiscent of a public search-engine. This requires working from web data, which is what users expect to have access to. In addition to precise answers, supporting passages must also be provided to convince the user of the validity of the answers. The reproducibility requirements preclude from working with live web data; a fixed but large collection was constituted instead.

Two languages were addressed: French and English. 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. Six types of questions were covered: factual, how, why, definition, list and boolean. The answer evaluation can take 6 statuses described in section 5.: *Full*, *Right*, *Unsupported*, *Supported*, *Inexact* and *False*. Three metrics were chosen for the primary evaluation: the Mean Reciprocal Rank (MRR), the First Hit Success (HS) or top-1, and the Hit Success (HS) or top-3. An answer is considered as correct according to these metrics if it is evaluated *Full* or *Right*.

3. The Data

3.1. The corpus

The document corpus for the QA evaluation has been created by taking the first 100 pages returned for the requests present in a set of web search logs. Exalead provided logs from their search-engine[2]. These extracted logs were in English and in French. A filtering was made to remove

[1] `http://www.quaero.org`

[2] `http://www.exalead.fr`

the non-pertinent (for the evaluation) requests such as sex or YellowPages. At the beginning 739,000 requests were available and after filtering 39,262 requests were kept for French, 9,820 for UK English and 29,442 requests for US English. Finally for each language the full corpus has a size of around 20Gb for 2.5M documents, from which a subset of around 500K documents was selected for the 2008 and 2009 evaluations. The table 1 gives the precise size for the French and English corpus actually used in the evaluations.

Table 1: corpus specifications

	French	English
Nb of documents	500 k	499 k
Nb of sentences	82 M	92 M
Nb of words	840 M	921 M
Nb of characters	4.2 G	4.9 G

3.2. Questions

Different types of questions were created for the evaluations. Factual, Non-factual, List, Boolean and Nil questions. The set of questions was created for each language by a native speaker. Table 2 shows the number of questions by type.

Table 2: Number of questions by type

	2008		2009	
	French	English	French	English
Nb questions	250	250	507	518
Factual	167	167	295	137
Non-factual	49	49	68	44
List	20	20	21	6
Boolean	6	6	28	14
nil	8	8	95	317
Nb of words	9.15	8.26	6.21	7.07

3.2.1. Questions for the 2008 evaluation

For the 2008 evaluation the question creation process was to first select a document by using a user request and afterwards to write a question by using the data in the document. For example for the request *wood manufacturing* the search-engine proposes a lot of links. By visiting the different pages a passage was found : "*Drew Graham, a wood manufacturing technician from Nova Scotia required a website to showcase his work for potential employers*". The question *Who is Drew Graham ?* was built from it. The associated answer was *wood manufacturing technician*.

This methodology guarantees an answer for each question. Nevertheless, the questions are close to the text and are not necessary close to a real application. The mean number of words per question was 9.15 for French and 8.26 for English. A set of 250 questions was created for this baseline evaluation, including 167 factual questions, 49 non factual questions, 20 list questions, 6 boolean questions and 8 *nil* questions. For each language a native speaker created the set of questions.

3.2.2. Questions for the 2009 evaluation

For the 2009 evaluation the questions were created without referring to the documents. A request was taken and a number of questions were generated from it when appropriate. Once again six types of questions were created for the evaluation. The simple factual, the complex factual (how, why, definition), the list and the boolean questions. Table 3 shows the number of created questions by type.

Table 3: Number of created questions by type for 2009 evaluation

	French	English
Nb questions	507	518
Factual Simple	350	353
Factual Complex	102	113
List	27	25
Boolean	28	27

In practice 800 requests were used from the logs to create those 1,025 questions.

Examples of question creation from requests are presented in table 4.

When creating the questions no information was available about whether an answer is present in the corpus. In fact, a large number of questions are left for which no correct answer has been found. The evaluator then checks the corpus by hand for answers using a simple search engine and his knowledge. Questions for which no answer is found (*nil* questions) are subsequently eliminated from the test. They comprised around 20% of the initial questions for French and 60% for English. Table 2 shows the final number of questions used in the 2009 evaluation. This large number of *nil* question is balanced with the fact that the questions seem to be closer to a real application. As mentioned in the table 2 the mean number of words used per question is less important than for 2008: that probably means that the questions are simpler and more general, perhaps more representative of what a human would ask a system.

4. Participants and systems

Four systems have participated to the evaluation. A description of each system follows.

4.1. The Ritel-QA system

The LIMSI Ritel-QA system has been built on the framework of the RITEL system (Schooten van et al., 2007). The RITEL project aims to integrate a spoken language dialogue system and an open-domain information retrieval system in order to enable human users to ask a general question and to refine interactively their search for information.

The same complete and multilevel analysis is carried out on both queries and documents. The general objective of this analysis is to find the bits of information that may be of use for search and extraction, called *pertinent information chunks*. These can be of different categories: named entities, linguistic entities (e.g., verbs, prepositions), or specific entities (e.g., scores, colors). All words that do not fall into such chunks are automatically grouped into chunks via a longest-match strategy. The full analysis comprises some

Table 4: Examples of question creation from request

request	question	type
london armoury company	*When was London Amoury Company founded?*	factual simple
most popular library blogs	*How is a blog created?* *What is a blog?*	how definition
cheap flights to paris	*Why are there so many tourists in Paris in April?*	why
cisco "clean access agent"	*Is CCAA the abbreviation for Cisco Clean Access Agent?*	boolean
captain john gower	*What battles did captain John Gower fight in?*	list

100 steps and takes roughly 4 ms on a typical user or document sentence. The analysis identifies about 300 different types of entities. The analysis is hierarchical, resulting in a set of trees. Both answers and important elements of the questions are supposed to be annotated as one of these entities (Rosset et al., 2008).

The results of the document analysis is stored in a specialized index. This index contains all pairs (type,value) produced by the analysis and provides the raw occurrence counts for each of the elements.

The first step of QA system itself is to build a search descriptor (SD) that contains the important elements of the question, and the possible answer types with associated weights. Some elements are marked as *critical*, which makes them mandatory in future steps, while others are *secondary*. The element extraction and weighting is based on an empirical classification of the element types in importance levels. Answer types are predicted through rules based on combinations of elements of the question.

Documents are selected using this SD. Each element of the document is scored with the geometric mean of the number of occurrences of all the SD elements that appear in it, and sorted by score, keeping the *n*-best. Snippets are extracted from the document using fixed-size windows and scored using the geometrical mean of the number of occurrences of all the SD elements that appear in the snippet, smoothed by the document score.

In each snippet, all the elements whose type is one of the predicted possible answer types are candidate answers. A score $S(r)$ is associated to each candidate answer *r*. This score is the sum of the the distances between the candidate answer and the elements of the SD, each elevated to the power $-\alpha$, ponderated by the element weights. That score is smoothed with the snippet score through a δ-ponderated geometric mean. All the scores for the different instances of the same element are added together, and in order to compensate for the differencing natural frequencies of the entities in the documents the final score is divided by the occurence count in all the documents and in all the examined snippets, each elevated to the power β and γ respectively. The entities with the best scores then win.

Moreover, for the *definition* questions we decided to use the analyzer in order to collect all possible definitions in the data collection and to store them in a table containing the definition itself, word entry and an identifier for the document in which the definition appears. For the *how* and *why* questions, we added two markers (*cause* and *manner*) in order to indicate when a procedure or an explanation is given in a document. For example,

le bleu du ciel <cause> est le résultat </cause> *de la diffusion de la lumière solaire par les composants de l' atmosphère.* (the blue of the sky is the result of the diffusion of the light by the components of the atmosphere)

peler les tomates <manner> en les plongeant </manner> *quelques secondes dans une casserole d' eau bouillante.* (peel the tomatoes by dunking them for a handful of seconds in boiling water)

These tags are useful only to extract a sentence in which the answer can be found.

For the 2009 QUAERO evaluation, we used the same system as for the 2008 evaluation but we added a new step for document selection. This new step works as a filter function which is based on language models. Roughly 30% of the previously selected documents are discarded. The other difference between the 2008 and 2009 systems concerns the analyser which has been improved.

4.2. The FIDJI system

FIDJI (Finding In Documents Justifications and Inferences), an open-domain QA system for French (Moriceau et al., 2009), combines syntactic information with traditional QA techniques such as named entity recognition and term weighting in order to validate answers. The main difficulty is that an answer (or some pieces of information composing an answer) may be validated by several documents. 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.

In this context, FIDJI's approach consists in checking if all the characteristics of a question (namely the dependency relations) may be retrieved in one or several documents. The 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]. Only a traditional bag-of-word indexing is necessary, and all fine linguistic analysis is performed online on a small subset of documents. First, the system submits the keywords of the question to Lucene: the top 100 documents are then processed (syntactic analysis and named entity tagging).

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

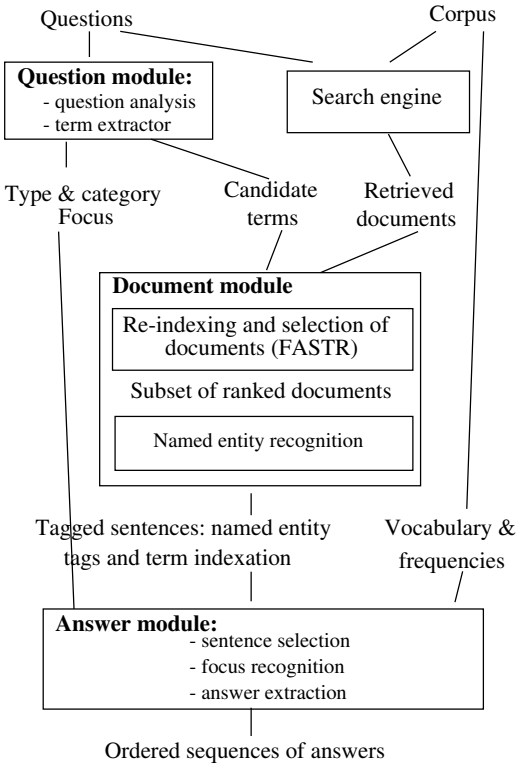

Figure 2: Architecture of QALC

Among these documents, FIDJI looks for all 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.

FIDJI was originally designed to process "clean" document collections (such as well-formed and syntactically correct news articles) and obtains good results (66% of correct answers on CLEF 2005) (Moriceau et al., 2009). For the Quaero evaluation, FIDJI has been adapted to process new question types.

4.3. The QALC system

The basic architecture of QALC (Ferret et al., 2002) is composed of different modules, one dedicated to the questions, one to the corpora, and a last module in charge of producing the answer. Each of these main modules is decomposed in several processes.

- Question module : the analysis of the questions allows to extract several pieces of information from the questions, among them:
  - an answer type that corresponds to the types of entities which are likely to constitute the answer to this question (a Named Entity type otherwise a concept type).
  - a question focus: a noun phrase that is the entity about which an answer is required and thus that is likely to be present in the answer
  - a question category that gives syntactic clues to locate the answer.
- Document module : the collection is indexed and searched by Lucene after a decomposition in overlapping units equivalent to a paragraph.
- Answer module : this module relies on two main operations: the sentence scoring and the answer extraction. All the data extracted from the questions and the documents by the preceding modules are used by a pairing module to evaluate the degree of similarity between a document sentence and a question. The answers are then extracted from the sentences according to several criteria:
  1. the presence of the expected answer type or not. In that experiment, this part was not used for constituting the final result with RITEL.
  2. the recognition of several question charasteristics in the sentence: focus, main verb, expected type when it is a concept type
  3. the category of the question and its associated patterns.

When the expected answer type is not a named entity, the QALC system locates the very answer within the candidate sentence through syntactic patterns. Syntactic patterns of answer include the focus noun phrase and the answer noun phrase, which can be connected by other elements such as comma, quotation marks, a preposition or even a verb. Thus, a syntactic pattern of an answer always includes the focus of the question. As a result, the focus has to be determined by the question analysis module in order to enable the QALC system to find a common noun or verb phrase as answer. If we consider the following question:
" What do Knight Ridder publish? "
The focus of the question, determined by the rules of the question analysis module, is *Knight Ridder*. This question pertains to the question type *What-do-NP-VB*, with *Knight Ridder* as NP and the verb *publish* as VB. One answer pattern applying to this category is called *FocusBeforeAnswerVB* and consists of the following syntactic sequence:
NPfocus Connecting-elements NPanswer
The *NPfocus* is the noun phrase corresponding to the question focus within the sentence-answer. It is followed by the connecting elements, then by a noun phrase that is supposed to contain the very answer. The connecting elements mainly consist of the question verb (VB in the question type). The following answer fits with the *FocusBeforeAnswerVB* pattern:
" Knight Ridder publishes 30 daily newspapers ... ",
This answer was extracted from the following sentence:
" Knight Ridder publishes 30 daily newspapers, including the Miami Herald and the Philadelphia Inquirer, owns and operates eight television stations and is a join venture partner in cable television and newsprint manufacturing operations. ".

In order to test a collaboration between the RITEL and the QALC systems, we did for this 2009 evaluation a mixed run. The aim was to apply the RITEL named entity detection processing to extract answers from the snippets selected by QALC. So, we first applied QALC. Then, for the

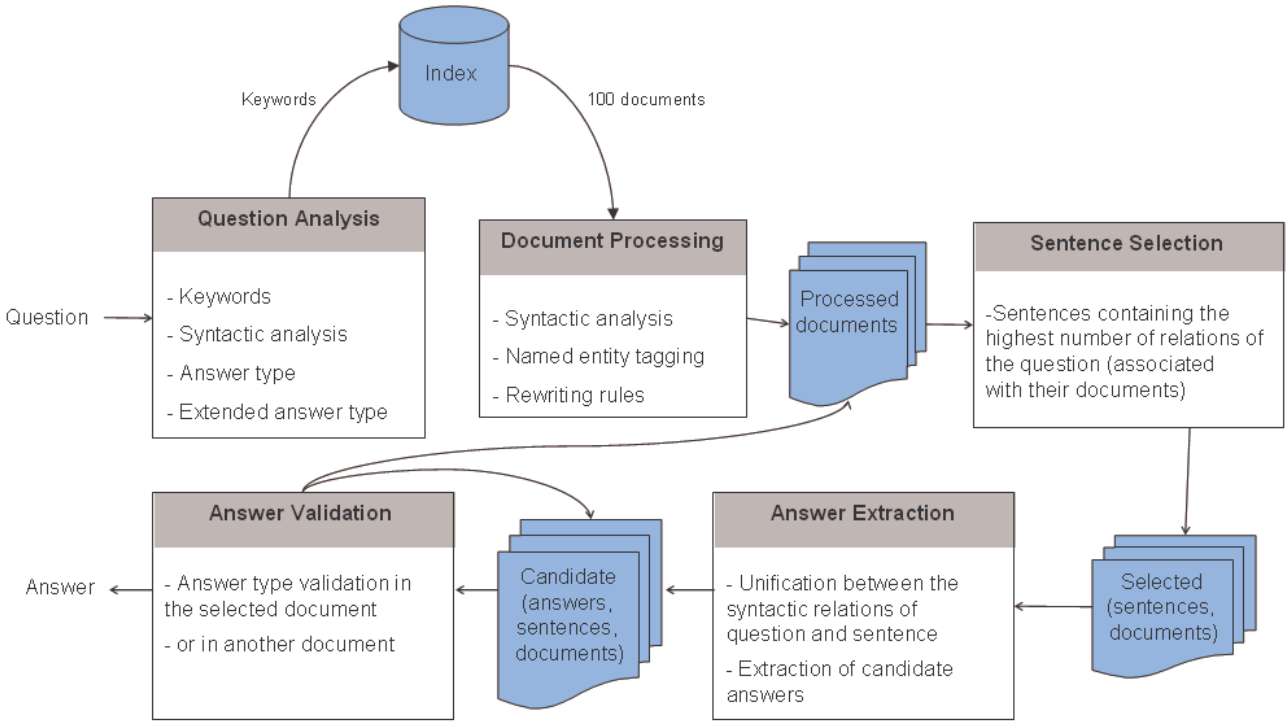

Figure 1: Architecture of FIDJI

questions that the QALC analysis has typed as expecting a named entity, we applied RITEL on the passages selected by QALC for these questions. Thus, the final run mixed the answers extracted by the two systems. The performance we obtained by this collaborative strategy is better than the current results of RITEL and the results

4.4. Synapse system

The systems from Synapse Développement are archetypal of the resource-heavy approach. Initially working in the grammatical correction field and since then diversifying into other fields such as question-answering, they developed a huge amount of resources describing the French and now the English language, including such things as a WordNet-equivalent, morphosyntaxic derivation tables, syntactic and semantic compatibility information, tables of expressions, etc. Using undisclosed algorithms leveraging these resources they can produce a complete, in-depth syntactic and semantic analysis of the text. Considering results of this analysis, 8 different indexes are built:

- heads of derivation. A head of derivation can be a sense for a word. In French, the verb voler has 2 different meanings (to steal or to fly). The meaning *dérober* (to steal) will lead to *vol* (robbery), *voleur* (thief) or *voleuse* (female thief). The second meaning, *se mouvoir dans l'air* (to fly), will lead to *vol* (flight), *volant* (flying as an adjective), *voleter* (to flutter) or *envol* (taking flight) and all its forms.

- proper names. If they appear in our dictionaries.

- idioms. Those idioms are listed in our idioms dictionaries. They encompass approximately 50,000 entries, like word processing, fly blind or as good as your word.

- named entities. Named entities are extracted from texts.*George W. Bush* or *Defense Advanced Research Project Agency* are named entities.

- concepts. Concepts are nodes of our general taxonomy. 2 levels of concepts are indexed. The first level lists 256 categories, like "visibility". The second level, actually the leaves of our taxonomy, lists 3387 subcategories, like "lighting" or "transparency",

- fields. 186 fields, like "aeronautics", "agriculture", etc.,

- question and answer types for categories like "distance", "speed", "definition", "causality", etc.,

- keywords of the text.

The question is syntactically and semantically analyzed by the system. After question analysis, all indexes are searched and the best ranked blocks are analyzed again. The analysis of the selected blocks is close to the analysis processed while indexing or question analyzing. On top of this "classic" analysis, a weight for each sentence is inferred. This weight is based on the number of words, synonyms and named entities found in this sentence, the presence of an answer corresponding to the question type and a correspondence between the fields and domain. After this analysis, sentences are ranked. Then, an additional analysis is processed to extract named entities, idioms or lists that match the answer. This extraction relies on the syntactic characteristics of those groups.

5. The Evaluation

5.1. Assessment

The submitted systems' outputs have been assessed by the evaluator by using a tool developped in Perl (at LNE). A simple interface enables easy access to the question, the answer (short and passage) and the document. The short answer is higlighted in the passage and in the document so that the accessor can see easily whether the answer is pertinent for the question. For each answer the assessor gives one of these 6 evaluations:

- A *Full* answer is a set formed by: first, a precise character string answering the question; second, a document justifying this answer; and third, a relevant quote from this document that proves the answer is correct.
- A *Right* answer is a precise character string which answers the question and is extracted from a document justifying it.
- An *Unsupported* answer is a precise character string which answers the question, but which is not extracted from a document justifying it.
- A *Supported* answer answer is a specific character string that does not provides an answer to the question but the passage is relevant.
- An *Inexact* answer is a not precise character string which answers the question and is extracted from a document justifying it.
- A *False* answer is a character string which does not answer the question.

5.2. The metrics

The computation of the final score was done by using the MRR, $top-3$ and $top-1$ metrics. The $top-3$ is used to know if for a question the system have a good answer in the first three answer. The $top-1$ permits to know if for a question the answer will be at the first rank. The $top-n$ can be computed as (1) :

$$\text{top-}n = \frac{\#\text{CR}_i \leq n}{\#\text{questions}} \quad (1)$$

The Mean Reciprocal Rank MRR gives information about the capacity of a system to catch a good answer in the first page. The Mean Reciprocal Rank can be computed as (2) :

$$MRR = \frac{\sum_{i=1}^{N_q} \frac{1}{rank_i}}{N_q} \quad (2)$$

where N_q is the number of questions and $rank_i$ is the rank of the correct answer used for the computation. A correct answer is an answer with the status Full or Right. The first correct answer was taken for the computation.
For the list, a comparison between the list provided by the system and a reference list is done. In 2008 the evaluation of the list is mesured by using this formula (3) :

$$Q = \max(0, \frac{C - (S - C)}{L}) \quad (3)$$

Table 5: Results for the 2008 and 2009 evaluation campaign for the factual simple question

French		
	2008	2009
MRR	0.196–0.426	0.284–0.540
top-3	0.288–0.472	0.393–0.579
top-1	0.159–0.386	0.275–0.502
English		
MRR	0.153–0.359	0.192–0.341
top-3	0.197–0.383	0.227–0.360
top-1	0.113–0.341	0.161–0.330

Table 6: Results for the 2008 evaluation campaign for the complex question

2008		
	French	English
MRR	0.196-0.426	0.047-0.108
top-3	0.086-0.217	0.047-0.108
top-1	0.043-0.195	0.040-0.102

When in 2009 we prefered to use the F-measure (4) :

$$P = \frac{C}{S} \qquad R = \frac{C}{L} \qquad F = \frac{2 \times P \times R}{P + R} \quad (4)$$

Where C is the number of common elements between the two lists. L is the number of elements in the reference list. S is the number of elements given by the system.

5.3. The results

For French the global MRR ranges from 0.254 to 0.433 and for English from 0.177 to 0.366, when in 2008 it ranges for French from 0.136 to 0.332 and for English from 0.123 to 0.289. In spite of the higher complexity due to the process used to build the questions the systems obtained better results. The English documents are much noisier than the French ones (spam, incorrectly detected language, etc) and it explains in the results. Table 5 gives a comparison between the results obtained during the 2008 and 2009 campaigns for the factual simple questions. We see in this table that the results on factual questions were 0.284-0.540 for French and 0.192-0.342 for English. Non-factual , harder ones got a result of 0.145-0.335 for French and 0.147-0.334 for English. In practice the yes/no questions were well addressed while the systems' outputs for the list questions were very poor. Even if results on French language are better than those obtained on English language, their ranges are comparable, and still less accurate than those obtained with journalistic corpus.
For the non factual questions the table 6 gives the results obtained in 2008. For 2009 the results are provided by table 7. For the 2009 evaluation these questions were divided into three type (How, Why, What).
The systems are better at answering the *definition* questions rather than the *How* and *Why* questions.
The table 8 gives results for the Boolean questions. Here only the MRR is given because the systems must answer only one time per question.

Table 7: Results for the 2009 evaluation campaign for the complex question(How, Why, definition)

How		
	French	English
MRR	0.088-0.490	0.051-0.153
top-3	0.117-0.588	0.153
top-1	0.058-0.411	0.000-0.153
Why		
	French	English
MRR	0.158-0.317	0.045-0.090
top-3	0.190-0.428	0.090
top-1	0.142-0.238	0.000-0.090
definition		
	French	English
MRR	0.166-0.261	0.476-0.380
top-3	0.166-0.333	0.476-0.380
top-1	0.166-0.200	0.047-0.380

Table 8: Results for the 2008 and 2009 evaluation campaign for the boolean question

French		
	2008	2009
MRR	0.0-0.333	0.208-0.827
English		
MRR	0.0-0.333	0.500-0.857

6. Conclusions & Future Work

The results of the Quæro 2009 evaluation are very encouraging, showing that QA technology is starting to be able to deal with complex and noisy corpus. All systems increased their results between 2008 and 2009 by about 40% relative. The task definition should be kept stable to be able to assess progress. Still, we will try to switch to the larger (20G) original corpus in order to, hopefully, benefit from a higher redundancy in the documents. Another way is to estimate the difference between the two corpora of questions. To do so, we will study the work done in (Bernard et al., 2010).
The corpus of questions with the answers and the snippets will be available in 2010. For any question about the corpus and how you can obtain it you may send an email at *ludovic.quintard@lne.fr* and *olivier.galibert@lne.fr*. The document corpus itself, being Web data, is not redistributable.

Acknowledgement

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

7. References

Christelle Ayache, Brigitte Grau, and Anne Vilnat. 2006. EQueR: the French Evaluation campaign of Question-Answering Systems. In *LREC 2006*, Genoa, Italy, May.

Salah Aït-Mokhtar, Jean Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: Incremental deep parsing. *Natural Language Engineering*, 8:121–144.

Guillaume Bernard, Sophie Rosset, Martine Adda-Decker, and Olivier Galibert. 2010. A question–answer distance measure to investigate QA system progress. In *LREC'10*.

Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Christian Jacquemin, Laura Monceaux, Isabelle Robba, and Anne Vilnat. 2002. How nlp can improve question answering. *Knowledge Organization*, 29(3-4).

Pamela Forner, Anselmo Peñas, Iñaki Alegria, Corina Forascu, Nicolas Moreau, Petya Osenova, Prokopis Prokopidis, Paulo Rocha, Bogdan Sacaleanu, Richard Sutcliffe, and Erik Tjong Kim Sang. 2008. Overview of the CLEF 2008 Multilingual Question Answering Track. In *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark, September.

Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji, and Noriko Kando. 2008. Overview of the NTCIR-7 ACLIA Tasks: Advanced Cross-Lingual Information Access. In *Proceedings of the Seventh NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, Tokyo, Japan.

Véronique Moriceau, Xavier Tannier, and Brigitte 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.

Sophie Rosset, Olivier Galibert, Guillaume Bernard, Eric Bilinski, and Gilles Adda. 2008. The limsi participation to the qast track. In *Working Notes of CLEF 2008 Workshop*, Aarhus, Denmark, September.

Boris Schooten van, Sophie Rosset, Olivier Galibert, Aurelien Max, Rieks Akker op den, and Illouz. 2007. Handling speech input in the ritel qa dialogue system. In *in InterSpeech'07*, Anvers, Belgique.

Ellen M. Voorhees and Donna K. Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing. September.