Selecting answers to questions from Web documents by a robust validation process

A. Grappy*†, B. Grau*‡, M-H. Falco*†, A-L. Ligozat*‡, I. Robba*§, A. Vilnat*†
*LIMSI (CNRS), †Université Paris-Sud, ‡ENSIIE, §UVSQ, France
firstName.Name@limsi.fr

***Abstract*—Question answering (QA) systems aim at finding answers to question posed in natural language using a collection of documents. When the collection is extracted from the Web, the structure and style of the texts are quite different from those of newspaper articles. We developed a QA system based on an answer validation process able to handle Web specificity. A large number of candidate answers are extracted from short passages in order to be validated according to question and passages characteristics. The validation module is based on a machine learning approach. It takes into account criteria characterizing both passage and answer relevance at surface, lexical, syntactic and semantic levels to deal with different types of texts. We present and compare results obtained for factual questions posed on a Web and on a newspaper collection. We show that our system outperforms a baseline by up to 48% in MRR.**

***Keywords*-fine-grained information retrieval; question-answering system; answer validation; Web document analysis;**

I. Introduction

The search for specific information in text, in response to factual questions posed in natural language, is an area widely studied since the first evaluation of open domain question-answering (QA) systems at TREC in 1999 for English, followed by CLEF campaigns for European languages, including French since 2005. Factual questions are questions that seek accurate information about an entity or an event, whatever the domain, as opposed to definition questions, opinion questions or complex questions such as *Why* or *How* questions. Best systems make use of deep Natural Language Processing (NLP) techniques, in order to match questions and candidate passages and extract answers [1], [2], [3] for some European languages and [4] for French. These systems require intensive handcrafted knowledge and cannot easily be adapted to new languages or new kinds of texts. Most other approaches developed more robust methods, based on calculations of similarity between questions and passages, as we did in our previous systems FRASQUES [5] for French and QALC [6] for English. The architecture commonly used in these systems consists in applying successive filters.

These different kinds of approaches proved to be quite successful on texts from newspaper articles or texts of the same type, with better performances on English texts than on French texts (as for systems on other European languages [7]), certainly due to the lack of available reliable resources. However they failed on the Quæro collection[1], made of documents extracted from the Web (cf. results in [8]). The systems reached an accuracy from 15.9% to 38.6% in the 2008 evaluation and from 27.5% to 50.2% in 2009 while the best French system at CLEF 2006 reached up to 68.95%.

When questioning the Web, systems have to deal with document structures specific to Web pages such as lists and tables, containing menus and navigation paths, etc. The textual information is not made of full and well written sentences, which poses problem for syntactic parsing.

Web pages show another specificity due to these structures: the distribution of the information provided by the question over several lines of texts. Few answers are given in a single sentence that allows one to assess their correctness. Justifying elements have to be searched in larger passages. Thus, we regarded answering questions as an answer validation problem on retrieved passages.

Answer validation was introduced at CLEF with AVE (Answer Validation Exercise[2]) in 2006. The aim of the AVE task was to automatically assess the validity of the answers given by QA systems [9]. The AVE task is close to Pascal Recognizing Textual Entailment Challenge[3] (RTE) that defines *"textual entailment"* as the task to decide, given two fragments of text, if the meaning of one can be deduced from the other [10], thus if a question plus a candidate answer (the hypothesis) can be deduced from a candidate passage. Passages provided in AVE corpus are the justifying excerpts of texts given by QA systems.

Few works have applied such a process in a complete QA system. They rely on different strategies for integrating an answer validation module in the system: for ordering passages and answers [11], for selecting between several sets of answers [12] or for finding answers over structured data modeled as a textual entailment problem between the query and semi structured patterns [13]. Validation methods rely mostly on machine learning approaches incorporating various criteria, most often of a lexical nature: terms of the hypothesis present in the passage, common named entities, presence of the expected type of answer, longest common chains [14], including dealing with linguistic variations

[1] http://www.quaero.org - Quæro is a program financed by OSEO, which partly financed this work, also supported by the CSOSG ANR project FILTRAR-S

[2] http://nlp.uned.es/QA/AVE/

[3] http://www.pascal-network.org/Challenges/RTE

between answer and question terms [15][16]. Besides, many works deal with passage or answer reranking, assuming they have lists of passages. They mainly studied how to account for syntactic and semantic correspondences between questions and passages, by computing similarities between their respective syntactic trees [17], or by computing common dependency paths [18]. They have proved to be useful on well written texts. However, such approaches relying on deep syntactic parsing cannot be applied on Web documents.

In order to deal with this specificity, we conceived a new system on French language, QAVAL (Question Answering by VALidation), that does not apply successive filters for selecting the right answer but extracts a lot of candidate answers from about 3-sentence passages provided directly by the search engine. Candidate answers are ranked by a validation process based on the characterization of an answer by different features given to a machine learning classifier.

The main characteristics of QAVAL are:

- preprocessing of the Web collection in order to provide parsable passages,
- justifications are searched in multi-sentence passages,
- only local syntax is considered to account for partial sentences,
- different strategies for validating answers according to the type of question and the type of information asked.

To overcome the difficulty posed by unstructured texts, our validation process is based on local features computed at different levels, including lexical, syntactic and semantic aspects as in [19].

To show the contribution of our validation process, we focused on finding answers to factual questions as pieces of information provided by this kind of questions are often split over several sentences. On such a base test, our system outperforms the baseline by up to 48% in MRR and obtains on French language state-of-the-art results of robust QA systems on English.

II. QAVAL SYSTEM

QAVAL is made of sequential modules, corresponding to five main steps (see Fig. 1). The question analysis provides main characteristics for retrieving passages and guiding the validation process. Short passages are obtained directly from the search engine and are annotated with question terms and their weighted variants. They are then syntactically parsed and enriched with the question characteristics, which allows QAVAL to compute the different features for validating or discarding candidate answers. We briefly present here the three first steps, while the two last ones, are detailed in the further sections.

A. Question analysis

This module aims at extracting all the elements useful for searching passages and extracting answers. It is based on the results provided by the syntactic parser XIP [20]. Questions are characterized by: i) a focus, i.e. the object about which an information is required; it is either an entity, referred by a noun phrase, or an event, referred by a verbal phrase, depending on the type of the question and the type of its main verb; ii) a question category, to classify the kind of relation that holds between the focus and the answer; iii) an expected type of answer, both a named entity type among general categories and a specific type when given in the question (as *animal* in *What animal* ...); iv) the main verb, if significant; v) the significant terms, either mono and multi-word units.

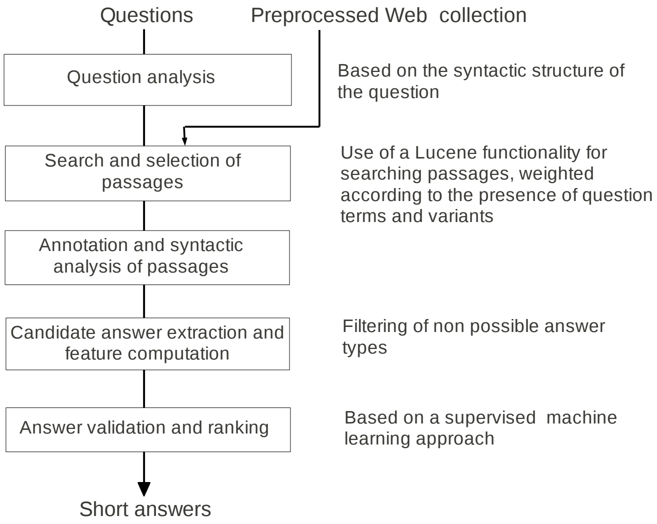
Questions
Preprocessed Web collection
Question analysis
Based on the syntactic structure of the question
Search and selection of passages
Use of a Lucene functionality for searching passages, weighted according to the presence of question terms and variants
Annotation and syntactic analysis of passages
Candidate answer extraction and feature computation
Filtering of non possible answer types
Answer validation and ranking
Based on a supervised machine learning approach
Short answers

Figure 1. The QAVAL system

B. Preprocessing for answer extraction: handling the absence of visual rendering

Documents were preprocessed to obtain homogeneous, usable (for syntactic parsing) and relevant documents. HTML documents were first converted into XHTML documents by sequentially applying *HTMLCleaner*[4] and *jTIDY*[5] for easier computational handling as a lot of them were not valid[6]. Then each relevant XHTML document was converted into a textual document where the textual content is segmented in units during its extraction. A linear extraction would produce segments of thousand words as visual space is often used for disposing rendered HTML content: a human will interpret a specific layout, for example an important vertical space, as a separation between two parts of text, even if there is no explicit ending symbol. This visual information is lost in the textual content (from HTML source code), and this is why a final punctuation mark was added during textual extraction if it did not exist like for example between

[4] http://htmlcleaner.sourceforge.net/

[5] http://jtidy.sourceforge.net/

[6] both regarding the W3C standards and the basic HTML source code

Table I
EXAMPLE OF A TABLE IDENTIFIED AS A DATA TABLE

Holidays	**Zone A**	**Zone B**	**Zone C**
Winter	9-02 25-02	2-02 18-02	16-02 4-03
Spring	6-04 22-04	30-03 15-04	13-04 29-04

two paragraphs or between a title (h1-h6) and a paragraph. On the opposite, split segments inside source code were joined whenever they seemed to belong to the same sentence but were separated by a carriage return: a linear extraction would separate them in two segments and leads the syntactic parser to make mistakes. The last step removed completely irrelevant documents like sitemaps as they only contain urls: although they are composed of useful segments of information, especially for document classification [21], they are too unlikely to contain the answer due to its formatting.

HTML structures like tables (see Table I) must not be linearly parsed. In this example, the italic cell is a topic cell, bold cells are headers and the others cells are data cells. A linear extraction would produce *Holidays Zone A Zone B Zone C Winter 9-02 25-02 2-02 18-02 16-02 4-03 Spring 6-04 22-04 30-03 15-04 13-04-29-04* while our extraction produces *Holidays ; Winter / Zone A / 9-02 25-02. Holidays ; Winter / Zone B / 2-02-18-02 (...) Holidays ; Spring / Zone A / 6-04-22-04. Holidays ; Spring / Zone B / 30-03-15-04 (...)*. The latter extraction will improve the results of the syntactic parser when recognizing phrases and is more likely to allow the extraction of the correct answer *30-03 15-04* for the question *When do Spring holidays occur for Zone B?* while the former would lead to extract (*6-04 22-04*), the closest date to *Spring*, which is incorrect.

The list tag was also preprocessed as there were too many lists composed of links (mainly menu or spam) so every item of a link list was deleted if it was not composed of at least five words once split on blank space (unlikely to be an analysable and useful segment).

C. Passage retrieval

In QAVAL, we chose to rely on the search engine to select a first set of passages that will be annotated with the mono and complex terms of the questions[7], along with their variants, and weighted according to the kinds of terms found (cf. section II-D). In a QA system, a passage is relevant if it is likely to contain the answer and to justify it, i.e. the passage holds the same information as the question.

The *Lucene* search engine[8] is used for indexation and retrieval with the following two features: stemming and use of snippets, for retrieving short fragments instead of document references. *Lucene* extracts a snippet by centering it on the continuous passage that matches mostly the query and allows to parametrize the maximal number of snippets retrieved by document, as there could be more than one fragment from the document that matches the query. If the number of passages is set to one, *Lucene* extracts the best snippet. It is also possible to parametrize the size of the snippet and past experimental results showed it was better to use a 300-character size in order to have nearly three sentences. A treatment to complete each retrieved snippet is done afterwards as the centering process generally cuts the first and last sentences.

D. Selection and annotation of relevant passages

The passages returned by Lucene are parsed by Fastr [22], a shallow parser which locates different kinds of variations of question terms (morphologic, syntactic or semantic variations). QAVAL gives a weight to each variation according to its reliability. Then, taking these weights into account, the best passages are selected [6]. A study conducted on a corpus of questions and their passages provided by *Lucene* showed that our weighting process ranked almost always a correct answer passage among the 50 first documents.

Then passages are annotated according to both their syntactic features and characteristics provided by question analysis (specific type, focus, main verb, name entity expected). All these elements are annotated step by step in order to facilitate the extraction of candidate answers. Numeric entities are annotated first. Then, using XIP, each sentence of the passage is parsed: the syntactic tree plus the set of dependency relations, which also contain named entities, are provided. In a last step, question characteristics are annotated: the focus and its modifiers if any, the specific type, the main verb. Wmatch [23] was used to this end; it applies regular expressions (whose basic unit is the lemma or the syntagm) to the syntactic trees of passage sentences.

III. EXTRACTION OF CANDIDATE ANSWERS

As the validation process is dedicated to assess the correctness of answers, our goal at this stage is to extract all the possible candidates. However, we cannot consider all the terms of a passage as candidates for computation time issue so we defined semantic and syntactic criteria to discard some of them. If we consider that factual questions ask for precisions about an entity or an event, answers can be restricted to the noun phrases that are modifiers of a noun or a verb (the *focus* in QAVAL). As this relation is not always marked in the passages, all nominal groups could be candidates. However their number remains too important, so we applied semantic criteria to restrict this set. Questions may expect a named entity as an answer (as *Who is the president of the United States?* which expects a person name in response) or not (as *Name a Michael Jackson's success.*), and are distinguished by the question analysis module.

For questions that expect a named entity in response, all named entities of the expected type annotated in the

[7] They are the significant words of the question, plus multiword units corresponding to noun phrase patterns

[8] http://lucene.apache.org/java/docs/index.html, version 2.9

passages are extracted, plus the proper nouns in unmarked noun phrases.

IV. Answer validation

The problem with the validation of candidate answers in QAVAL is slightly different from the AVE task. The answers to validate are much less relevant than in AVE where they had been previously selected by QA systems. Another difference concerns the justifying passages: QAVAL passages are larger than AVE passages and extracted from Web pages and not from newspapers. At first, QAVAL rejects what it considers as incorrect answers. Then, features are computed and given to a classifier to rank the remaining candidates.

A. Candidate filtering

As many candidate answers are extracted, a first step consists in recognizing obvious false answers. Answers from a passage that does not contain all the named entities of the question are eliminated. Rodrigo et al. [24] show that this method rejects few correct answers. In QAVAL, this criterion is applied to passages that do not contain a *person*, an *organization*, a *date* or a *location* present in the question, which are the most reliable named entities. For example, for the question *Which country invaded Kuwait in 1990?*, the passages which do not contain *1990* and *Kuweit* do not contain a valid answer. Concerning the answers, those which are fully present in the question are removed; for example the answer *Jackson* to the question *Name a Michael Jackson's success?* is clearly an incorrect answer.

The remaining answers are ranked by a learning method. Two types of features are used: features characterizing passages and features characterizing answers.

B. Features relative to the passage

To be relevant for providing and justifying an answer, a passage has to convey the same meaning as the question. The following features are designed to assess this property: they compare the words of the question and those of the passage, assuming that they have to share same syntactic relations.

These hypotheses lead to a first set of features:

- number of significant words[9] of the question in the passage; it accounts for same lemmas or recognized variations;
- words by category; the importance of a word of the question varies according to its morpho-syntactic category: a proper name is more important than an adjective. The features are the proportions of proper names, common nouns, verbs, adjectives and numerical expressions of the question present in the passage;
- important words: the question analysis extracts important words (focus, specific type, main verb). The presence of these words in the passage is captured in features;
- multi-word units;
- passage rank: a score is computed for selecting passages after their annotation by question terms;

The second set of features characterizes the answer, by itself or in relation with the passage.

C. Features relative to the candidate answer

An answer has to be of an expected type, if explicitly required and to be related to the question terms, and specially to the focus. Another kind of criterion concerns the answer redundancy: the most frequent an answer is, the most relevant it is.

1) Proximity of the terms: If an answer is close to the question words then they are probably connected. To evaluate this hypothesis, two features are computed.

The first one computes the longest common chain of consecutive words [14] of the passage and of the question rewritten in declarative form completed by the candidate answer. Two words are consecutive if they are adjacent or separated by stop words and possibly a bonus word. The feature is the proportion of words common to the question and the passage present in the substring.

The second feature represents the average distance between the answer and each of the question word. If a question word is not in the passage then the distance is the length of the passage. With this feature, the smaller the value is, the closer to question words the answer is and the most relevant the answer could be.

2) Question category: The question analysis distinguishes four categories of questions, depending on the relation of the answer with question terms (focus or type), and its possible lexicalization: noun modifier, verb subject or object, verb modifier, unit number.

3) Type checking: Numerous questions wait for answers of an explicit type. For example, the question *Which president succeeded Jacques Chirac?* requires an instance of *president* as an answer. We apply a machine learning method to verifying that a candidate answer corresponds to the required type. Features are based on statistical criteria and computed from the presence rate of both the answer and the type in documents. Another feature is based on the presence of the candidate answer in a knowledge base in which entities are associated to fine-grained named entity types (as *movie, river, ...*).

Wikipedia pages are used to compute the last kinds of features, as the content of a page generally defines its title. For example, the word *physicist* is in the page corresponding to *Albert Einstein*. Thus, a feature denotes the presence of the expected type in the page corresponding to the answer and another is set when sentence structures indicating the presence of a definition between the answer and the type like *Albert Einstein is a physicist* are found in the pages.

[9] nouns, adjectives, verbs that do not belong to a stoplist

Theses features are combined by a decision tree. The method is presented in more detail in [25]. Its evaluation obtains a F-measure of 0.80.

4) Redundancy: If the same answer is extracted from many texts it is more likely to be a correct answer. The associated feature is the number of instances of this answer.

D. Learning method

The combination of the preceding features is learned by a combination of decision trees[10], as in [11], which recursively select the best feature for separating data.

The classifier provides a confidence score between -1 and 1 to each answer, indicating its confidence in the validity of the answer. The value 1 indicates that the answer is correct and -1 that it is invalid. This score allows us to rank the different answers in order to obtain the most reliable results in first position. It is to be noted that for an answer with different instances, only the most reliable is kept.

V. Experimentation

A. The experimentation frame

The corpus was rawly[11] crawled by the Exalead[12] company in May and June 2008: user's queries logged into the Exalead web search-engine had been used to collect referenced documents into a collection of nearly two million documents. From this collection, Exalead extracted a representative subset of 500,000 documents that we have been using in evaluations[13] for computational reasons.

Documents crawled had to include textual content. Nothing was done for HTML and XML documents but a textual conversion had been applied to non-HTML documents by Exalead: textual contents were linearly extracted into XML files either with a very basic structure (notably by page for DOC and PDF documents) or without any structure (SWF).

Two experiments were conducted for evaluating the performance of our system and its robustness. Test sets of questions are based on past evaluations on French in which we only kept the factual questions.

- EQUER, based on the CLEF collection. QAVAL searches for the answer of 126 questions from the EQUER evaluation [26] in newspapers documents.
- Quæro based on the Quæro Kitten collection. 147 questions coming from the Quæro 2010 campaign were used to test our system. For those questions the Web documents are searched.

All questions have an answer in the collections; 150 passages are retrieved by *Lucene* and 50 passages are selected afterwards.

The factual questions from QA@CLEF05 and QA@CLEF06 were used to create the training set, by running QAVAL. A set of triplets (question, answer patterns, passage) are proposed to a manual evaluation to reject correct answer strings that are not explained by the passage. Two training sets were built. For the first one, QAVAL looks at the answer in a collection of newspaper articles from *Le Monde* and ATS newswires (the CLEF collection). The second set comes from the application of QAVAL on the Quæro Web collection. The first training set contains 950 valid answers and 1900 invalid answers, the second 349 valid answers and 698 invalid answers. The number of incorrect answers was reduced to correspond to $\frac{2}{3}$ of the data, originally there were 53781 answers.

MRR (*Mean Reciprocal Rank*) and accuracy are used to evaluate the results. We computed the MRR on the top five answers, and the accuracy, that measures the number of questions that are correctly answered, is computed at the first rank and at the top five ranks. An answer is seen as correct if it is equal to an answer pattern which have been manually collected.

B. Evaluation of the selected passages

We evaluated the passage extraction from the HTML Quæro collection by counting the ratio of questions for which at least one of the retrieved snippets contains the correct answer. We also computed the MRR on the final answers.

The metric is applied (see Table II) to the passages returned by *Lucene* from the collection extracted with our preprocessing, named Quæro Kitten, with the Quæro collection extracted with Boilerpipe[14] [27] and with the baseline which consists in a linear textual extraction of tags having textual content (without any preprocessing or selection).

Table II
PASSAGE EXTRACTION EVALUATION

	# correct retrieved snippet	QAVAL results (MRR)
Kitten	**130 (88 %)**	**0.43**
BoilerPipe	121 (82 %)	0.32
baseline	114 (77 %)	0.28

The results show that *Boilerpipe* is not optimal for a Web collection: extracting only the most dense textual fragment is a good strategy for newspaper collection, but, on the Quæro collection, data are too sparse and not structured enough to avoid loosing important fragments.

For characterizing the Quæro Kitten collection compared to a newspaper collection, we counted the number of sentences in the parsed passages before the answer extraction

[10] bagging method in WEKA : http://sourceforge.net/projects/weka/

[11] only the online file from the referenced url was crawled: not CSS, nor DTD or any other online file used by this online file

[12] http://www.exalead.com/software/

[13] Evaluation data are only available to the participants to the Quæro project

[14] http://code.google.com/p/boilerpipe/

process (see Table III). We can see that the Quæro passages are made of shorter sentences that are twice as many as in CLEF collection, and that these sentences contain half the number of verbs. These values characterize the types of structures found in Web pages: successions of noun phrases due to tables, lists, etc.

Table III
AVERAGE NUMBER OF SENTENCES PER PASSAGE

	Avg nb sentences	**Avg. nb. verbal sentences**	**Avg length**	**Verb ratio**
EQUER	2.70	2.56	27.87	11.76
Quæro	5.82	2.32	12.37	8.3

C. Answer validation results

We compared our method to a baseline method for selecting and ranking answers. This method extracts the candidate closest to the question words from each top five passages, in their ranking order. Table IV presents the results of QAVAL on the two test sets along with the baseline scores. It also presents the evaluation of the answer validation method by itself by computing the scores according to the number of remaining questions that still can be answered when the validation process occurs in the QA system.

Table IV
QAVAL RESULTS

QAVAL evaluation	MRR	First rank %(#)	Top five ranks %(#)
EQUER	**0.47**	**39% (49)**	**60% (76)**
Quæro	**0.43**	**34% (50)**	**56% (82)**
baseline EQUER	0.34	27% (34)	47% (59)
baseline Quæro	0.29	21% (32)	43% (64)

Answer validation evaluation	MRR	First rank	Top five ranks
EQUER (113 questions)	**0.53**	**43%**	**67%**
Quæro (122 questions)	**0.49**	**40%**	**66%**
baseline EQUER	0.38	30%	52%
baseline Quæro	0.36	28%	49%

QAVAL outperforms the baseline on Quæro up to 48% which shows the appropriateness of our answer selection method and its efficiency on the two kinds of documents as the distance between the two baselines is maintained, even if the task remains more difficult on the Web documents. The validation method obtains good results with 66% (resp. 67%) of the questions having an answer in the five first ranks and 40% (resp. 43%) at the first rank.

Figure 2 illustrates where answers are lost, with the accuracy obtained at each step of the system: passage extraction, passage selection, answer extraction and answer validation.

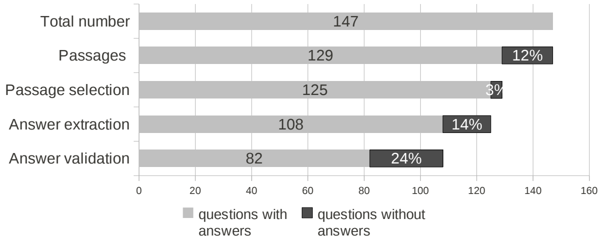

Figure 2. Error repartition

We can see that a good passage extraction and selection are realized: only 15% of questions are not associated to a correct passage anymore. 24% of questions having at least an answer correctly extracted do not have a right answer in the five first ranks anymore. The high number of candidates may explain this result: 11,405 candidates have to be validated (77 candidates by question in average). The same remarks can be observed on the EQUER test.

To evaluate the features roles in the validation module, we analyzed the different decision trees and we saw that all features are used. We tested the impact of those criteria that are specific to QAVAL: the validation of the answer type and the question category. QAVAL obtains a MRR of 0.41 on the QUAERO questions without the type verification, versus 0.43 with it, which shows the importance of this feature although it is applied to 55 % of the questions. Same report was done concerning the utilility of the question category.

Figure 3 presents how answer features improve text features on the two experiments Quæro and EQUER. We can see that for EQUER answer frequency and term proximity are two important features and not the verification of the answer type. Contrarily, in Quæro, the answer type verification is more important than the answer frequency. This confirms the difference of the two corpus.

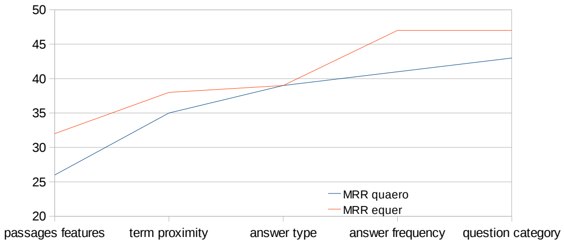

Figure 3. Feature importance

To evaluate the answer validation module in Quæro, we computed the MRR on the passages as they are ranked before its application : we obtained 0.36. We then computed the MRR of the passages containing the top five answers after the validation step, obtaining 0.54, which improves the passage reranking up to 50%. We also evaluated the accuracy of the first passage (i.e. the passage which contains the first answer proposed) which is of 0.58, and can be compared to 0.39, the accuracy of QAVAL. This result underlines that other criteria related to the characterization of the correct answer had to be determined.

Tests were conducted to evaluate the extraction method on the EQUER corpus. The first test consists in extracting all noun phrases as candidates for questions expecting a *person* or an *organization* name as answer. This test highlights the importance of the named entity filter because the MRR drops from 0.47 to 0.39. The second test evaluates the filter that leads to reject wrong answers. Without this module the MRR also drops to 0.39.

At the answer selection step, we separate questions depending on the fact that they expect a named entity or not as an answer. Results are better if a question expects a named entity (MRR 0.52 vs 0.25 on EQUER questions) as an answer. That can be explained by the higher number of candidate answers extracted for questions without expected named entity. We can venture the hypothesis that finding reliable constraints to reduce the number of extracted answers will improve our results.

Others systems described in [8] obtain a MRR between 0.284 and 0.54 when looking for the answer to factual questions on the Quæro corpus; QAVAL results are rather close to the best results.

VI. Related work

The task of answer validation (AVE) [9] at CLEF was dedicated to validate answers to questions in relation with a justification passage, both provided by QA systems. RTE purpose is to decide if a passage entails a hypothesis, that can be equivalent to a question in its declarative form completed by an answer.

Structured approaches aim at comparing a structured representation of the question plus the answer and the passage, based either on syntactic representations and an edit distance [17], [28] or on a semantic inference chain as COGEX [29] that performs the validation by a logical proof, proceeding by refutation. These kinds of approach can apply on well formed texts that can be successfully parsed.

Other systems rely mostly on machine learning approaches incorporating various criteria, most often of a lexical nature: terms of the hypothesis present in the passage, common named entities or similarity measures. To get a better fit when comparing terms, systems make use of external semantic knowledge such as WordNet for English. A criterion frequently used is the longest common substring between the question and the passage [14], that also may reflect linguistic variations [16], [15]. Such a criterion allows systems to take into account both syntactic and lexical similarities in a same measure, with common words and common syntactic roles, considering that if the hypothesis and the passage share an important subpart, there is a strong evidence that their topic are similar. However, criteria based on syntactic dependencies can also be explicitly introduced as a criterion [16]. Works on answer or passage ranking deal with same problem of ranking. Many systems, [30], [18] are based on a learning method that makes use of lexical features like presence of question terms in passage, term proximity or syntactic features. The evaluation conditions in Cui et al. [18] are very close to ours, except for the kind of texts, so we can compare our results. Our validation process reaches 0.49 and 0.53 on the top five ranks vs 0.4761 and an accuracy of 0.38 and 0.43 vs 0.3889. Ittycheriah et al. [31] improves his question answering system by reranking the answers using a maximum entropy method (from 0.458 to 0.496).

Harabagiu et al.[11] added entailment methods to a question answering system. Text entailment consists in a learning method based on common syntactic dependencies, a lexical alignment of semantic annotations and a paraphrase module. The entailment method computes a confidence score used to rank passages during the passage selection and to rank answers for the final ranking. Using these two methods improved the results of 20%.

VII. Conclusion

Most QA systems lack of robustness when they search Web documents: the structure of the documents (with a lot of lists, tables, etc.), and the quality of the textual content prevent these systems to apply processes developed with success on newspaper or encyclopedic documents. To deal with such documents, we develop a QA system QAVAL based on answer validation to choose the most relevant answer to a given question. Candidate answers are extracted from short passages without applying restrictive selection criteria. The task to select the answers is devoted to the answer validation module, based on learning methods. Several different features are taken into account in this process, concerning the passage in which the answer is selected or the answer itself. QAVAL obtains good results, outperforming a baseline by up to 48% in MRR. We show that our results are quite similar on newspaper documents and on Web documents, proving the robustness of QAVAL. Moreover QAVAL and the validation process present state-of-the-art results, even compared with systems ranking *well written passages*. To improve those results, we plan to take into account new criteria, leading to a better ranking. One possible criterion could be to make use of a paraphrase module able to verify that the action described in a question is paraphrased in the passage.

The validation process was developed on French language, however, to be adapted to other languages, it requires a syntactic parser, a lexicon with word variations, and needs to adapt the analysis of questions.

References

[1] A. Hickl, J. Williams, J. Bensley, K. Roberts, Y. Shi, and B. Rink, "Question answering with LCC's CHAUCER at TREC 2006," in *Proceedings of the Fifteenth Text REtrieval Conference*, 2006.

[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 spécial Répondre es questions*, vol. 46, no. 3, 2005.

[3] S. Hartrumpf, "University of Hagen at QA@ CLEF 2005: Extending knowledge and deepening linguistic processing for question answering," in *Working Notes, CLEF*, 2005.

[4] D. Laurent, P. Séguéla, and S. Nègre, "Cross lingual question answering using qristal for clef 2006," *Evaluation of Multilingual and Multi-modal Information Retrieval*, 2010.

[5] B. Grau, A.-L. Ligozat, I. Robba, A. Vilnat, and L. Monceaux, "FRASQUES: A Question Answering system in the EQueR evaluation campaign," in *Language Resources and Evaluation*, 2006.

[6] O. Ferret, B. Grau, M. Hurault-plantet, G. Illouz, and C. Jacquemin, "Terminological variants for document selection and question/answer," in *matching, ACL 2001 Workshop on Open-Domain Question Answering*, 2001.

[7] B. Magnini, D. Giampiccolo, P. Forner, C. Ayache, V. Jijkoun, P. Osenova, A. Peñas, P. Rocha, B. Sacaleanu, and R. F. E. Sutcliffe, "Overview of the CLEF 2006 Multilingual Question Answering Track," in *Working Notes for CLEF*, 2006.

[8] L. Quintard, O. Galibert, G. Adda, B. Grau, D. Laurent, V. Moriceau, S. Rosset, X. Tannier, and A. Vilnat, "Question Answering on web data: the QA evaluation in Quæro," in *Proceedings of LREC'10*, 2010.

[9] A. Peñas, Á. Rodrigo, V. Sama, and F. Verdejo, "Overview of the answer validation exercise 2006," *Evaluation of Multilingual and Multi-modal Information Retrieval*, 2010.

[10] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor, "The second pascal recognising textual entailment challenge," in *The Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

[11] S. Harabagiu and A. Hickl, "Methods for using textual entailment in open-domain question answering," in *Proceedings of the 44th annual meeting of ACL*, 2006.

[12] A. Téllez-Valero, M. Montes-y Gómez, L. Villaseñor-Pineda, L. del Lenguaje, and A. Peñas-Padilla, "Towards Multi-Stream Question Answering Using Answer Validation," *Informatica*, vol. 34, pp. 45–54, 2010.

[13] M. Negri, M. Kouylekov, and B. Magnini, "Detecting Expected Answer Relations through Textual Entailment," in *Computational Linguistics and Intelligent Text Processing*. Springer Berlin / Heidelberg, 2008.

[14] E. Newman, N. Stokes, J. Dunnion, and J. Carthy, "UCD IIRG Approach to the Textual Entailment Challenge," in *Proceedings of the PASCAL Challenges Workshop on RTE*, 2005.

[15] A.-L. Ligozat, B. Grau, A. Vilnat, I. Robba, and A. Grappy, "Lexical validation of answers in question answering," in *International Conference on Web Intelligence*, 2007.

[16] A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi, "Recognizing Textual Entailment with LCC's GROUNDHOG System," in *Proceedings of the PASCAL Challenges Workshop on RTE*, 2006.

[17] M. Kouylekov, M. Negri, B. Magnini, and B. Coppola, "Towards Entailment-based Question Answering: ITC-irst at CLEF 2006," in *CLEF 2006*, 2006.

[18] H. Cui, R. Sun, K. Li, M. yen Kan, and T. seng Chua, "Question answering passage retrieval using dependency relations," in *SIGIR 2005*, 2005.

[19] J. Ko, E. Nyberg, and L. Si, "A probabilistic graphical model for joint answer ranking in question answering," in *Proceedings of the 30th annual international SIGIR conference*, 2007.

[20] S. Aït-Mokhtar, J.-P. Chanod, and C. Roux, "Robustness beyond shallowness: incremental deep parsing," *Natural Language Engineering*, vol. 8, 2002.

[21] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," in *ACM Computing Surveys*, 2009.

[22] C. Jacquemin, "Syntagmatic and paradigmatic representations of term variation," in *Proceedings of the 37th annual meeting of ACL*, 1999.

[23] S. Rosset, O. Galibert, G. Bernard, E. Bilinski, and G. Adda, "The LIMSI participation to the QAst track," in *Working Notes of CLEF 2008 Workshop*, 2008.

[24] Á. Rodrigo, A. Peñas, J. Herrera, and F. Verdejo, "The Effect of Entity Recognition on Answer Validation," in *CLEF 2006*, 2006.

[25] A. Grappy and B. Grau, "Answer type validation in question answering systems," in *RIAO*, 2010.

[26] C. Ayache, B. Grau, and A. Vilnat, "EQueR : the French Evaluation Campaign of Questions Answering Systems," in *Language Resources and Evaluation (LREC'10)*, 2006.

[27] C. Kohlschütter, P. Fankhauser, and W. Nejdl, in *WSDM*, B. D. Davison, T. Suel, N. Craswell, and B. Liu, Eds. ACM, pp. 441–450.

[28] O. Ferrandez, D. Micol, R. Munoz, and M. Palomar, "The contribution of the University of Alicante to AVE 2007," in *Working Notes of CLEF*, 2007.

[29] I. Glöckner, "University of Hagen at QA@CLEF 2006: Answer Validation Exercise," in *Working Notes for the CLEF 2006 Workshop*, 2006.

[30] J. Suzuki, Y. Sasaki, and E. Maeda, "SVM answer selection for open-domain question answering," in *COLING*, 2002.

[31] A. I. Martin, M. Franz, and S. Roukos, "IBM's Statistical Question Answering System-TREC-10," in *In Proceedings of TREC10*, 2001.