Two levels of evaluation in a complex NL system

Jean-Baptiste Berthelin
LIMSI-CNRS
Bât.508 Université Paris XI
91403 Orsay, France
jbb@limsi.fr

Brigitte Grau
LIMSI-CNRS
Bât.508 Université Paris XI
91403 Orsay, France
bg@limsi.fr

Martine Hurault-Plantet
LIMSI-CNRS
Bât.508 Université Paris XI
91403 Orsay, France
mhp@limsi.fr

Abstract

The QALC question-answering system, developed at LIMSI, has been a participant for two years in the QA track of the TREC conference. In this paper, we present a quantitative evaluation of various modules in our system, based on two criteria: first, the numbers of documents containing the correct answer and selected by the system; secondly, the number of answers found. The first criterion is used for evaluating locally the modules in the system, which contribute in selecting documents that are likely to contain the answer. The second one provides a global evaluation of the system. As such, it also serves for an indirect evaluation of various modules.

1 Introduction

For two years, the TREC Evaluation Conference, (Text REtrieval Conference) has been featuring a Question Answering track, in addition to those already existing. This track involves searching for answers to a list of questions, within a collection of documents provided by NIST, the conference organizer. Questions are factual or encyclopaedic, while documents are newspaper articles. The TREC9-QA track, for instance, proposed 700 questions whose answers should be retrieved in a corpus of about one million documents.

In addition to the evaluation, by human judges, of their systems' results (Voorhees and Tice, 2000), TREC participants are also provided with an automated evaluation tool, along with a database. These data consist of a list of judgements of all results sent in by all participants. The evaluation tool automatically delivers a score to a set of answers given by a system to a set of questions. This score is derived from the mean reciprocal rank of the first five answers. For each question, the first correct answers get a mark in reverse proportion to their rank. Those evaluation tool and data are quite useful, since it gives us a way of appreciating what happens when modifying our system to improve it.

We have been taking part to TREC for two years, with the QALC question-answering system (Ferret et al, 2000), currently developed at LIMSI. This system has following architecture: parsing of the question to find the expected type of the answer, selection of a subset of documents among the approximately one million TREC-provided items, tagging of named entities within the documents, and, finally, search for possible answers. Some of the components serve to enrich both questions and documents, by adding system-readable data into them. Such is the case for the modules that parse questions and tag documents. Other components operate a selection among documents, using added data. One example of such modules are those which select relevant documents, another is the one which extracts the answer from the documents.

A global evaluation of the system is based on judgement about its answers. This criterion provides only indirect evaluation of each component, via the evolution of the final score when this component is modified. To get a closer evaluation of our modules, we need other criteria. In particular, concerning the evaluation

of components for document selection, we adopted an additional criterion about selected relevant documents, that is, those that yield the correct answer.

This paper describes a quantitative evaluation of various modules in our system, based on two criteria: first, the number of selected relevant documents, and secondly, the number of found answers. The first criterion is used for evaluating locally the modules in the system, which contribute in selecting documents that are likely to contain the answer. The second one provides a global evaluation of the system. It also serves for an indirect evaluation of various modules.

2 System architecture

Figure 1 shows the architecture of the QALC system, made of five separate modules: Question analysis, Search engine, Re-indexing and selection of documents, Named entity recognition, and Question/sentence pairing.

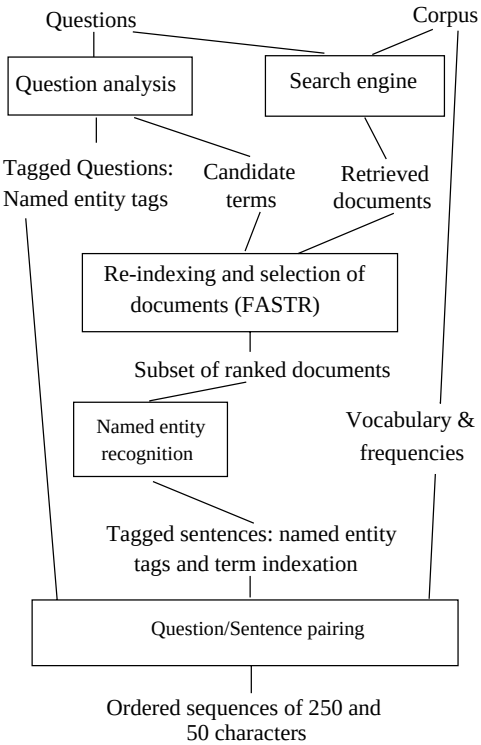

Figure 1. The QALC system

2.1 Question analysis

Question analysis is performed in order to assign features to questions and use these features for the matching measurement between a question and potential answer sentences. It relies on a shallow parser which spots discriminating patterns and assigns categories to a question. The categories correspond to the types of named entities that are likely to constitute the answer to this question. Named entities receive one of the following types: person, organisation, location (city or place), number (a time expression or a number expression). For example the pattern *how far* yields to the answer type *length*:

Question: How far away is the moon?
Answer type: LENGTH
Answer within the document :
With a <b_numex_TYPE="NUMBER"> 28 <e_numex> -power telescope you can see it on the moon <b_numex_TYPE="LENGTH"> 250,000 miles <e_numex> away.

2.2 Selection of relevant documents

The second module is a classic search engine, giving, for each question, a ranked list of documents, each of which could contain the answer.

This set of documents is then processed by a third module, made of FASTR (Jacquemin, 1999), a shallow transformational natural language analyser and of a ranker. This module can select, among documents found by the search engine, a subset that satisfies more refined criteria. FASTR improves things because it indexes documents with a set of terms, including not only the (simple or compound) words of the initial question, but also their morphological, syntactic and semantic variants. Each index is given a weight all the higher as it is close to the original word in the question, or as it is significant. For instance, original terms are considered more reliable than semantic variants, and proper names are considered more significant than nouns. Then, documents are ordered according to the number and the quality of the terms they contain. An analysis of the weight graph of the indexed documents enables the system to select a relevant subpart of those documents, whose size varies along the questions. Thus, when the curve presents a high negative slope, the system only select documents before the fall, otherwise a fixed threshold is used.

2.3 Named entity recognition

The fourth module tags named entities in documents selected by the third one. Named entities are recognized through a combination of lexico-syntactic patterns and significantly large lexical data. The three lists used for lexical lookup are CELEX (1998), a lexicon of 160,595 inflected words with associated lemma and syntactic category, a list of 8,070 first names (6,763 of which are from the CLR (1998) archive) and a list of 211,587 family names also from the CLR archive.

2.4 Question-sentence pairing

The fifth module evaluates each sentence in the ranker-selected documents, using a similarity measure between, on one side, terms and named entities in the sentence, and on the other side, words in the questions and expected answer type. To do so, it uses the results of the question parser, and the named entity tagger, along with a frequency-weighted vocabulary of the TREC corpus.

The QALC system proposes long and short answers. Concerning the short ones, the system focuses on parts of sentences that contain the expected named entity tags, when they are available, or on the larger subpart without any terms.

3 Search engine evaluation

The second module of the QALC system deals with the selection, through a search engine, of documents that may contain an answer to a given question from the whole TREC corpus (whose size is about 3 gigabytes).

We tested three search engines with the 200 questions that were proposed at the TREC8 QA track. The first one is Zprise, a vectorial search engine developed by NIST. The second is Indexal (de Loupy et al 1998), a pseudo-boolean search engine developed by Bertin Technologies[1]. The third search engine is ATT whose results to the TREC questions are provided by NIST in the form of ranked lists of the top 1000 documents retrieved for each question. We based our search engine tests on the list of relevant documents extracted from the list of correct answers provided by TREC organizers.

Since a search engine produces a large ranked list of relevant documents, we had to define the number of documents to retain for further processing. Indeed, having too many documents leads to a question processing time that is too long, but conversely, having too few documents reduces the possibility of obtaining the correct answer. The other goal of the tests obviously was to determine the best search engine, that is to say the one that gives the highest number of relevant documents.

3.1 Document selection threshold

In order to determine the best selection threshold, we carried out four different tests with the Zprise search engine. We ran Zprise for the 200 questions and then compared the number of relevant documents respectively in the top 50, 100, 200, and 500 retrieved documents. Table 1 shows the test results.

Selection Threshold	Questions *with* relevant documents	Questions *with no* relevant documents
50	181	19
100	184	16
200	193	7
500	194	6

Table 1. Number of questions with and without relevant documents retrieved for different thresholds

According to Table 1, the improvement of the search engine results tends to decrease beyond the threshold of 200 documents. The top 200 ranked documents thus seem to offer the best trade-off between the number of documents in which the answer may be found and the question processing time.

3.2 Evaluation

We compared the results given by the three search engines for a threshold of 200 documents. Table 2 gives the tests results.

[1] We are grateful to Bertin Technologies for providing us with the outputs of Indexal on the TREC collection for the TREC8-QA and TREC9-QA question set.

Search Engine	Indexal	Zprise	ATT
Number of questions *with* relevant documents retrieved	182	193	194
Number of questions *without* relevant documents retrieved	18	7	6
Total number of relevant documents that were retrieved	814	931	1021

Table 2. Compared performances of the Indexal, Zprise and ATT search engines

All three search engines perform quite well. Nevertheless, the ATT search engine revealed itself the most efficient according to the following two criteria: the lowest number of questions for which no relevant document was retrieved, and the most relevant documents retrieved for all the 200 questions. Both criteria are important. First, it is most essential to obtain relevant documents for as many questions as possible. But the number of relevant documents for each question also counts, since having more sentences containing the answer implies a greater probability to actually find it.

4 Document ranking evaluation

As the processing of 200 documents by the following Natural Language Processing (NLP) modules still was too time-consuming, we needed an additional stronger selection. The selection of relevant documents performed by the re-indexing and selection module relies on an NLP-based indexing composed of both single-word and phrase indices, and linguistic links between the occurrences and the original terms. The original terms are extracted from the questions. The tool used for extracting text sequences that correspond to occurrences or variants of these terms is FASTR (Jacquemin, 1999). The ranking of the documents relies on a weighted combination of the terms and variants extracted from the documents. The use of multi-words and variants for document weighting makes a finer ranking possible.

The principle of the selection is the following: when there is a sharp drop of the documents weight curve after a given rank, we keep only those documents which occur before the drop. Otherwise, we arbitrarily keep the first 100.

In order to evaluate the efficiency of the ranking process, we proceeded to several measures. First, we apply our system on the material given for the TREC8 evaluation, one time with the ranking process, and another time without this process. 200 documents were retained for each of the 200 questions. The system was scored by 0.463 in the first case, and by 0.452 in the second case. These results show that document selection slightly improves the final score while much reducing the amount of text to process.

However, a second measurement gave us more details about how things are improved. Indeed, when we compare the list of relevant documents selected by the search engine with the list of ranker-selected ones, we find that the ranker loses relevant documents. For thirteen questions among the 200 in the test, the ranker did not consider relevant documents selected by the search engine. What happens is: the global score improves, because found answers rank higher, but the number of found answers remains the same.

The interest to perform such a selection is also illustrated by the results given Table 3, computed on the TREC9 results.

Number of documents selected by ranking	100	<<100
Distribution among the questions	342 (50%)	340 (50%)
Number of correct answers	175 (51%)	200 (59%)
Number of correct answer at rank 1	88 (50%)	128 (64%)

Table 3. Evaluation of the ranking process

We see that the selection process discards documents for 50% of the questions: 340 questions are processed from less than 100 documents. For those 340 questions, the average number of selected documents is 37. The document set retrieved for those questions has a weight curve with a sharp drop. QALC finds more often the correct answer and in a better position for these 340 questions than for the 342 remaining ones. These results are very interesting when applying such time-consuming processes as named-entities recognition and

question/sentence matching. Document selection will also enable us to apply further sentence syntactic analysis.

5 Question-sentence pairing evaluation

We sent to TREC9 two runs which gave answers of 250 characters length, and one run which gave answers of 50 characters length. The first and the last runs used ATT as search engine, and the second one, Indexal. Results are consistent with our previous analysis (see Section 3.2). Indeed, the run with ATT search engine gives slightly better results (0.407 strict)[2] than those obtained with the Indexal search engine (0.375 strict). Table 4 sums up the number of answers found by our two runs.

Rank of the correct answer retrieved	Run using ATT	Run using Indexal
1	216	187
2 to 5	159	185
Total of correct answers retrieved	375	372
No correct answer retrieved	307	310

Table 4. Number of correct answers retrieved, by rank, for the two runs at 250 characters

The score of the run with answers of 50 characters length was not encouraging, amounting only 0.178, with 183 correct answers retrieved[3].

5.1 Long answers

From results of the evaluation concerning document ranking, we see that the performance level of the question-sentence matcher depends partly on the set of sentences it has parsed, and not only on the presence, or absence, of the answer within these sentences. In other words, we do not find the answer each time it is in the set of selected sentences, but we find it easily if there are few documents (and then few sentences) selected. That is because similarity assessment relies upon a small number of criteria, which are found to be insufficiently discriminant. Therefore, several sentences obtain the same mark, in which case, the rank of the correct answer depends on the order in which sentences are encountered.

This is something we cannot yet manage, so we evaluated the matcher's performance, without any regard to the side effect induced by document processing order. As remarked in 3.2, search engines perform well. In particular, ATT retains relevant documents, namely, those that yield good answers, for 97 percent of the questions. The ranker, while improving the final score, loses some questions. After it stepped in, the system retains relevant documents for 90% of the questions. The matcher finds a relevant document in the first five answers for 74% of the questions, but answers only 62% of them correctly. Finding the right document is but one step, knowing where to look inside it is no obvious task.

5.2 Short answers

A short answer is selectively extracted from a long one. We submitted this short answer selector (under 50 characters) to evaluation looking for the impact of the expected answer type. Among TREC questions, some expect an answer consisting of a named entity: for instance a date, a personal or business name. In such cases, assigning a type to the answer is rather simple, although it implies the need of a good named entity recognizer. Answers to other questions (*why* questions for instance, or some sort of *what* questions), however, will consist of a noun or sentence. Finding its type is more complex, and is not done very often.

Some systems, like FALCON (Harabagiu et al 2000) use Wordnet word class hierarchies to assign types to answers. Among 682 answers in TREC9, 57.5% were analysed by our system as named-entity questions, while others received no type assignment. Among answers from our best 250-character run, 62.7% were about named entities. However, our run for shorter answers, yielding a more modest score, gives 84% of named-entities answers. In our system answer type assignment is of surprisingly small import, where longer answers are concerned. However, it does modify the selecting process,

[2] With this score, the QALC system was ranked 6th among 25 participants at TREC9 QA task for answers with 250 characters length.

[3] With this score, the QALC system was ranked 19th among 24 participants at TREC9 QA task for answers with 50 characters length.

when the answer is extracted from a longer sentence.

Such evaluations help us to see more clearly where our next efforts should be directed. Having more criteria in the similarity measurement would, in particular, be a source of improvement.

6 Discussion

We presented quantitative evaluations. But since we feel that evaluations should contribute to improvements of the system, more qualitative and local ones also appear interesting.

TREC organizers send us, along with run results, statistics about how many runs found the correct answer, and at which rank. Such statistics are useful in many ways. Particularly, they provide a characterisation of *a posteriori* difficult questions. Knowing that a question is a difficult one is certainly relevant when trying to answer it. Concerning this problem, de Loupy and Bellot (2000) proposed an interesting set of criteria to recognize *a priori* difficult questions. They use word frequency, multi-words, polysemy (a source of noise) and synonymy (a source of silence). They argue that an "intelligent" system could even insist that a question be rephrased when it is too difficult. While their approach is indeed quite promising, we consider that their notion of *a priori* difficulty should be complemented by the notion of *a posteriori* difficulty we mentioned: the two upcoming examples of queries show that a question may seem harmless at first sight, even using de Loupy and Bellot's criteria, and still create problems for most systems.

From these statistics, we also found disparities between our system and others for certain questions. At times, it finds a good answer where most others fail and obviously the reverse also happens. This is the case in the two following examples. The first one concerns an interesting issue in a QA system that is the determination of which terms from the question are to be selected for the question-answer pairing. This is particularly important when the question has few words. For instance, to the question *How far away is the moon?*, our term extractor kept not only *moon (NN)*, but also *away (RB)*. Moreover, our question parser knows that *how far* is an interrogative phrase yielding a LENGTH type for the answer. This leads our system to retrieve the correct answer: *With a 28-power telescope, you can see it on the moon 250,000 miles away*[4].

The second example concerns the relative weight of the terms within the question. When a proper noun is present, it must be found in the answer, hence an important weight for it. Look at the question *Who manufactures the software, « PhotoShop »?*. The term extractor kept *software(NN)*, *PhotoShop(NP)*, and *manufacture(VBZ)* as terms to be matched, but the matcher assigns equal weights to them, so we could not find the answer[5]. Later, we modified these weights, and the problem was solved.

Indeed, evaluation corpus seems to be difficult to build. Apart from the problem of the question difficulty level, question type distribution may also vary from a corpus to another. For instance, we note that TREC8 proposed much more questions with named entity answer type (about 80%) than TREC9 (about 60%). Thus, some participants who trains their systems on the TREC8 corpus were somehow disapointed by their results at TREC9 with regards with their training results (Scott and Gaizauskas, 2000).

However, it is generally hard to predict what will happen if we modify the system. A local improvement can result in a loss of performance for other contexts. Although the system's complexity cannot be reduced to just two levels (a local one and a global one), this can be an efficient step in the design of improvements to the whole system via local adjustments. But this is a very frequent situation in engineering tasks.

7 Conclusion and perspectives

Each evaluation reflects a viewpoint, underlying the criterion we use. In our case, the choice of criteria was guided by the existence of two main stages in the QA process, namely the selection of relevant documents and the selection of the answer among the selected documents sentences. Sometimes, such criteria concur in

[4] Among the 42 runs using 250 byte limit, submitted at TREC9-QA, only seven found the correct answer at rank 1, and 27 do not found it.

[5] 22 runs, out of 42 found the right answer at rank 1. Only 9 were unable to find it.

revealing the same positive or negative feature of the system. They can also yield a more precise assessment of the reasons behind these features, as was the case in our evaluation of the ranker. Moreover, when a system consists of several modules, their specific evaluations should imply different criteria.

This is particularly true in dialogue systems, where different kinds of processes are co-operating. Since information retrieval is an interactive task, it seems natural to associate a dialogue component to it. Indeed, users tend to ask a question, evaluate the answer, and reformulate their question to make it more specific (or, contrariwise, more general, or quite different). A QA system is, therefore, a good applicative setting for a dialogue module. Quantitative assessment of the QA system would be useful in assessing the dialogue system in this particular context. Such a global assessment would provide an objective judgement about whether the task (finding the answer) was achieved, or not. Successfulness in a task is a necessary component of the evaluation, nevertheless it is just a part of it. Obviously, dialogue evaluation is also a matter of cost (time, number of exchanges) and of user-friendliness (cognitive ergonomy).

However, objectivity is almost impossible to attain in these domains. In a recent debate (LREC 2000), serious objections about natural language tools evaluation and validation were developed e.g. by Sabah (2000). The main issue he raises is about the great complexity of such systems. However, we consider that by going as far as possible in the experimental search for evaluation criteria, we also make a meaningful contribution to this debate. While it is true that complexity should never be ignored, we consider that, by successive approximate modelisation and evaluation cycles, we can capture some of it at each step of our system's developement.

References

CELEX. 1998. http://www.ldc.upenn.edu/readme_files/celex.readme.html. Consortium for Lexical Resources, UPenns, Eds.

CLR. 1998. http://crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat#D3. Consortium for Lexical Resources, NMSUs, Eds., New Mexico.

Fabre C., Jacquemin C, 2000. Boosting variant recognition with light semantics. Proceedings COLING 2000, pp. 264-270, Luxemburg.

Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. Cambridge, MA, MIT Press.

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Jacquemin C. (2000), QALC — the Question-Answering system of LIMSI-CNRS. Pre-proceedings of TREC9, NIST, Gaithersburg, CA.

Harabagiu S., Pasca M., Maiorano J. 2000. Experiments with Open-Domain Textual Question Answering. Proceedings of Coling'2000, Saarbrucken, Germany.

Jacquemin C. 1999. Syntagmatic and paradigmatic representations of term variation. Proceedings of ACL'99. 341-348.

de Loupy C., Bellot P., El-Bèze M., Marteau P.-F.. Query Expansion and Classification of Retrieved Documents, *TREC7* (1998), 382-389.

de Loupy C., Bellot P. 2000. Evaluation of Document Retrieval Systems and Query Difficulty. Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000) Workshop, Athens, Greece. 32-39.

Sabah G. 2000 To validate or not to validate? Some theoretical difficulties for a scientific evaluation of natural language processing systems. Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000) Workshop, Athens, Greece. 58-61.

Scott S., Gaizauskas R. 2000. University of Sheffield TREC-9 Q & A System. Pre-proceedings of TREC9, NIST, Gaithersburg, CA. 548-557.

Voorhees E., Tice D. 2000. Implementing a Question Answering Evaluation. Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000). Athens, Greece. 40-45.