The LIMSI participation to the QAst track

Sophie Rosset, Olivier Galibert, Guillaume Bernard, Eric Bilinski, Gilles Adda,
Spoken Language Processing Group, LIMSI-CNRS, B.P. 133, 91403 Orsay cedex, France
{firstname.lastname}@limsi.fr

Résumé

In this paper, we present the LIMSI question-answering systems on speech transcripts which participated to the QAst 2008 evaluation. These systems are based on a complete and multi-level analysis of both queries and documents. These systems use an automatically generated research descriptor. A score based on those descriptors is used to select documents and snippets. The extraction and scoring of candidate answers is based on proximity measurements within the research descriptor elements and a number of secondary factors. We participated to all the subtasks and submitted 18 runs (for 16 sub-tasks). The evaluation results for manual transcripts range from 31% to 45% for accuracy depending on the task and from 16 to 41% for automatic transcripts.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing ; H.3.3 Information Search and Retrieval ; H.3.4 Systems and Software ; H.3.7 Digital Libraries

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, speech transcriptions

1 Introduction

In the QA and Information Retrieval domains progress has been demonstrated via evaluation campaigns for both open domain and limited domains [7, 4, 1]. In these evaluations systems are presented with either independent or linked questions and should provide one answer extracted from textual data to each question. Recently, there has been growing interest in extracting information from multimedia data such as meetings, lectures... Spoken data is different from textual data in various ways. The grammatical structure of spontaneous speech is quite different from written discourse and include various types of disfluencies. The lecture and interactive meeting data provided in QAst evaluation are particularly difficult due to run-on sentences and interruptions. Most of the QA systems use a complete and deep syntactic and semantic analysis of both the question and the document, or snippets given by a search engine, and search for the answer in the result. Such an analysis cannot be performed reliably on the data we are interested in.

The Question Answering on Speech Transcripts track of the QA@CLEF task gives then an opportunity to evaluate approaches able to handle speech transcriptions.

In this paper, we present the architecture of the QA systems developed at LIMSI for the QAst evaluation. This year 10 general subtasks have been proposed :

- T1a : Question Answering in manual transcriptions of lectures (CHIL corpus)
- T1b : Question Answering in automatic transcriptions of lectures (CHIL corpus)
- T2a : Question Answering in manual transcriptions of meetings (AMI corpus)
- T2b : Question Answering in automatic transcriptions of meetings (AMI corpus)
- T3a : Question Answering in manual transcriptions of broadcast news for French (ESTER corpus)
- T3b : Question Answering in automatic transcriptions of broadcast news for French (ESTER corpus)
- T4a : Question Answering in manual transcriptions of European Parliament Plenary sessions in English (EPPS English corpus)
- T4b : Question Answering in automatic transcriptions of European Parliament Plenary sessions in English (EPPS English corpus)
- T5a : Question Answering in manual transcriptions of European Parliament Plenary sessions in Spanish (EPPS Spanish corpus)
- T5b : Question Answering in automatic transcriptions of European Parliament Plenary in Spanish (EPPS Spanish corpus)

For the tasks T3b, T4b and T5b, 3 different collections (one collection corresponding to one automatic speech recognition output) have been provided with 3 different Word Error Rates (WER) in order to allow studies on the impact of the WER on the Question Answering task. We submitted 2 runs for T3a and T5a tasks and one for each other tasks. In total, we submitted 18 runs. We used the exact same system for each manual and ASR collection in order to be able to evaluate the impact of the WER on the overall system. For the different languages and tasks, we used basically the same system, the only changes were the analysis which is language dependant and the tuning parameters learned on the development data set.

Figure 1 shows the general organisation of the system.

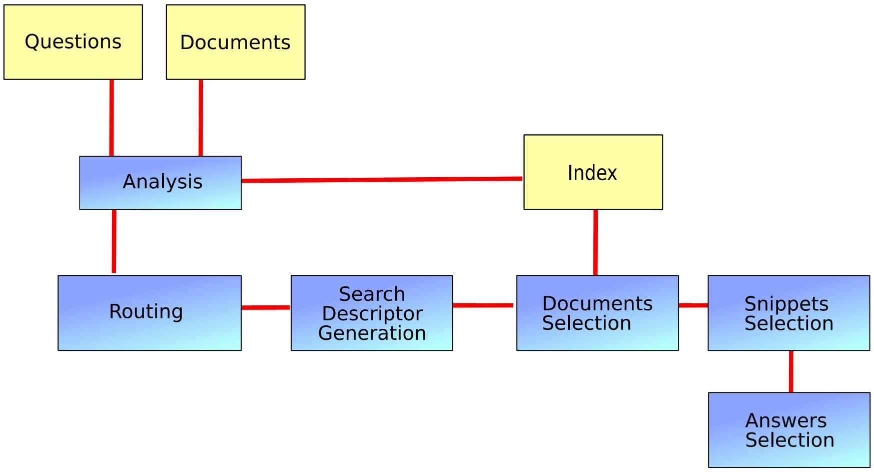

FIG. 1 – General overview of the LIMSI QAst systems

The following sections present the documents and queries pre-processing and the non-contextual analysis with the work carried out this year on the adaptation of our analysis system to Spanish. In section 3, we present the documents and snippets selection and the answer extraction and scoring. Section 4 finally presents the results for these two systems on both development and test data.

2 Analysis of documents and queries

Usually, the syntactic/semantic analysis is different for the document and for the query ; our approach is instead to perform the same complete and multilevel analysis on both queries and documents. There are several reasons for this : First of all, the system has to deal with both transcribed speech (transcriptions of meetings and lectures, user utterances) and text documents, so there should be a common analysis that takes into account the specifics of both data types. Moreover, incorrect analysis due to the lack of context or limitations of hand-coded rules are likely to happen on both data types, so using the same strategy for document and utterance analysis helps to reduce their negative impact. But first, we need to reduce the surface forms variations between the different modalities (text, manual transcripts, automatic transcripts) in order to have a common representation and use of words, sentences, case, etc. This process, a superset of tokenization, is called normalization.

2.1 Normalization

Normalization, in our application, is the process by which *raw* texts are converted to a text form where words and numbers are unambiguously delimited, capitalization happens on proper nouns only, punctuation is separated from words, and the text is split into sentence-like segments (or as close to sentences as is reasonably possible). Different normalization steps are applied, depending of the kind of input data ; these steps are :

1. Separating words and numbers from punctuation.
2. Reconstructing correct case for the words.
3. Adding punctuation.
4. Splitting into sentences at period marks.

Reconstructing the case and adding punctuation is done in the same process based on using a fully-cased, punctuated language model [3]. A word graph was built covering all the possible variants (all possible punctuations added between words, all possible word cases), and a 4-gram language model was used to select the most probable hypothesis. The language model was estimated on House of Commons Daily Debates, final edition of the European Parliament Proceedings and various newspapers archives. The final result, with uppercase only on proper nouns and words clearly separated by white-spaces, is then passed to the non-contextual analysis.

2.2 Analysis module

The *non-contextual analysis* aims at extracting, from both user utterances and documents, what is considered to be *pertinent information*. The analysis covers multiple levels : Named entities detection, Linguistic chunking, Question words classification and Question topic detection. An example of an analysis result appears on figure 2. In that example, *New-York* is recognized as a named entity, specifically an organization. *municipal elections* is chunked together as a compound noun, which makes it available as a search key in the QA system. *who* is detection as a question word related to a person, and its combination with *won* allows to classify the question as one about someone's victory or achievement.

The types we need to detect correspond to two levels of analysis : named-entity recognition and chunk-based shallow parsing. Various strategies for named-entity recognition using machine learning techniques

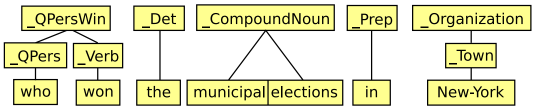

FIG. 2 – Example of user utterance analysis

have been proposed [2, 5, 6]. In these approaches, a statistically pertinent coverage of all defined types and subtypes induced the need of a large number of occurrences, and therefore rely on the availability of large annotated corpora which are difficult to build. Rule-based approaches to named-entity recognition (e.g. [8]) rely on morphosyntactic and/or syntactic analysis of the documents. However, in the present work, performing this sort of analysis is not feasible : the speech transcriptions are too noisy to allow for both accurate and robust linguistic analysis based on typical rules. We use a internal tool to write grammars based on regular expressions on words. Our tools allows the use of lists for initial detection, and the definition of local contexts and simple categorizations. This engine matches (and substitutes) regular expressions using words as the base unit instead of characters. This property allows for a more readable syntax than traditional regular expressions and enables the use of classes (lists of words) and macros (sub-expressions in-line in a larger expression).

2.2.1 Adaptation to English and Spanish languages

This analysis is obviously language dependant. The French analyser detects about 300 types and constitutes the basis for the Spanish and English (T4 task only) analyzers adaptation. This year was our first attempt in working with spanish. The Spanish analyser has been created as a simple adaptation of the French one where only the lexicons were adapted, and only around 50% of them. For the English a deeper adaptation is required, in particular the order in which the blocks of rules are applied is reversed. The English and Spanish analysers detect only about a hundred types.

We now plan to use some aligned corpus in order to automatically acquire some specific lexicons.

3 Question-Answering system

The input request takes the form of an analyzed question. From that information a *Search Descriptor* is built which is the basis of all the following search algorithms.

3.1 Search Descriptor Generation

This descriptor is structured in 3 parts : the elements of the input considered pertinent for the search, the expected type or types for the answer, and a number of tuning parameters.

The types considered pertinent are the named entities (standard, extended and nonspecific) and the linguistic chunks. Each entity also carries a weight, set by rules, and a critical/secondary flag. Critical entities must be present in a document near a candidate answer, secondary entities only give a bonus to the final score. This distinction aims at increasing the system precision. In practice, all named entities and some linguistic chunks are considered critical according to, once again, a set of rules. The expected answer types and their weights are decided using a 2-level rule-based classifier built by examining the development data and generalized by hand. Rhe tuning parameters are set empirically by systematic trials on the development data. Moreover, as shown in Figure3, possible transformations of the elements are described. These possible transformations are obtained from a few rules. This year, we used this concept to allow weighted morphological derivations and synonymic transformations. The lexicon used for morphological derivations have been built on our corpus using the analysis module to extract all values of the considered types (for example all adjectives and nouns) and to apply some derivational rules on these lists in order to built morphological correspondances. We tried various algorithms and that simple method was the one obtaining the best results on the development data set for each language and task.

Question : *when was Hans Krasa killed ?*

- Critical element
  - 1,0 *pers* identity(Hans Krasa)
  - 0,2 *pers* expand(Hans Krasa)
- Secondary element
  - 1,0 *verb* identity(killed)
  - 0,7 *verb* lemma(killed)
  - 0,5 *verb* synonym(killed)
  - 0,5 *subs* verb_subs(killed)
- Answer types
  - 1,0 *full_date*
  - 0,9 *month_year*, *day_month*, *hour*
  - 0,7 *year*

FIG. 3 – Example of a Search Descriptor : each element contains the list of triplets (type, transformation, value) under which it is expected to appear. Each triplet is weighted (*0,5* verb *synonym(killed)* a synonym of *killed* is accepted with a weight of 0.5) ; each possible answer type contains also a weight.

3.2 Documents selection and scoring

Once the Search Descriptor (SD) is built, the next step is to generate a list of the n documents with the highest probability of containing the answer. The method is fundamentally simple : give a score to all the documents that include at least one element of the SD and pick the n with the best scores. The score we've chosen is based on the counts of occurrences of elements, ponderated by the SD weights. The tree structure is taken into account : the scores of elements in the same node are added, the scores for children have their geometric mean taken. The geometric mean has two advantages, it avoids needing to compensate for the differences in global frequency of the elements, since the counts are multiplied together, and it ensures that a zero count on a critical element propagates into a global zero count. Accordingly, 1 is added to the secondary element nodes to avoid the zero-propagation effect. The document score is the score of a virtual root node of all the top nodes.

The index gives the raw occurrence counts for each of the elements. The analysis producing hierarchical annotations, the same instance of an elements can appear under multiple types. For instance, France is typed as both country and location or organization each time it appears in a document. To compensate for that the counts are recomputed by subtracting the number of occurrences taken into account for the other elements of the same or upper nodes.

In the specific case of QAst where the document count is very low, n is set high enough that all the documents with as least one element are picked.

3.3 Snippets selection and scoring

The snippet selection step aims at selecting in the documents blocks of lines with a high expectation of containing the answer. That action has a dual effect : faster answers by reducing the number of candidates to look at, and better precision of the answers given by reducing the noise introduced by faraway candidates.

The idea of the method is that elements of the SD has a *distance of influence* or *range* which is counted in lines, that is sentences for text documents or utterances for spoken documents. The algorithm starts by extracting all the lines which have elements in range to satisfy all the critical elements of the SD, building that way a series of blocks. Too big blocks, i.e. above a critical *size*, are split up to try to push them under the critical size by temporarily promoting some of the secondary elements to critical status. Eventually all the blocks are small enough or all the elements have become critical and no more splitting is possible.

We want these snippets to be self-contained for later candidate evaluation, which means that they must include all the elements found in the SD that made them pertinent. But in some cases two critical elements are too far apart from each other that the line they're in is kept, while some lines in the middle are within range of both and as such form an element-less snippet. To fix these situations the snippets frontiers are extended to cover the neighboring lines where influential elements are present.

3.4 Answers selection and scoring

The snippets are sorted by score and examined one by one independently. Every element in a snippet with a type found in the list of expected answer types of the SD is considered an answer candidate. Each candidate is given a score, which is the sum of the the distances between itself 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. This extraction and scoring stops once a number m of candidates has been reached, once again to control the speed of the system. 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. The tuning parameters α, β, γ, δ all come from the third part of the SD and has been selected by systematic trials on the develoment corpus. These parameters are set for each question class.

Our second approach for answer scoring is built upon the results of that first one. We compute a new ranking of the answers with a tree transformation method. For each candidate answer to a question, we transform the tree of the snippet from where the answer was extracted into the tree of the question. The sequence of operations used for the transformation gives us a transformation cost. The candidate answers are re-ranked using these costs. We applied this method as a second run for T3a and T5a tasks. The results do not yet show the expected improvement. But this work is still in progress and further analysis is needed. One positive aspect of these trials is that they show that this approach is completely language independant (same results are obtained for French and Spanish languages).

4 Evaluation

4.1 Training and Development data

The official development data consisted of 50 questions for each task. The development documents were 10 seminars for the T1 task, 50 meetings for the T2 task, 6 shows for the T3 task, 4 for the T4 task and 1 for the T5 task. As we have observed last year, 50 questions are clearly not enough to correctly tune a system. We decided to hand-build and use a corpus of reformulated questions for each task and used them as training corpus. We built corpus of questions/answering/documents for the T3, T4 and T5 tasks and we used the 2007 evaluation data for T1 and T2 tasks as blind development data. The table 1 gave a general overview of the different corpus used.

	Off. Dev.	Ref. q.	Blind Corpus
T1	50 (10)	565 (10)	100 (15)
T2	50 (50)	587 (50)	100 (118)
T3	50 (6)	350 (6)	248 (3 new)
T4	50 (3)	277 (3)	186 (6)
T5	50 (1)	217 (1)	36 (1 + 1 new)

TAB. 1 – The corpus : *Off. Dev.* : the official development data ; *Ref. q.* : the reformulated questions based on the development documents ; *Blind Corpus* : 2007 test data for T1 and T2 and new questions for T4, new questions and new documents for T3 and T5 ; Between parenthesis is the number of documents

4.2 Results

4.2.1 General results on manual transcripts

We compared the results obtained on our different corpus (training, on which the tuning is done, and development, blind corpus on which only the synthetic scores are looked at) and on the 2008 evaluation. The following tables give results obtained on the different development sets and on the test.

	50	50+	bc	test
Accuracy	96%	83.5%	64.3%	41%
MRR	0.98	0.85	0.71	0.45
Recall	100%	88%	80.6%	52%

TAB. 2 – **T1a task (English, seminar data)** : Results on the 3 different corpus and the test (*50* : 50 official development questions ; *50+* : 565 reformulated questions ; *bc* : 2007 test data ; *test* : 2008 test)

	50	50+	bc	test
Accuracy	66%	60.4%	44.8%	33%
MRR	0.72	0.66	0.52	0.40
Recall	82%	75.5%	61.5%	51%

TAB. 3 – **T2a task (English, meeting data)** : Results on the 3 different corpus and the test (*50* : 50 official development questions ; *50+* : 587 reformulated questions ; : 2007 test data ; *test* : 2008 test)

	50	50+	bc	test
Accuracy	82%	79.1%	41.5%	45%
MRR	0.90	0.86	0.50	0.49
Recall	100%	94.9%	61.3%	58%

TAB. 4 – **T3a task (French, BN data** : Results on the 3 different corpus and the test (*50* : 50 official development questions ; *50+* : 350 reformulated questions ; *bc* : Blind Corpus, 248 questions on 3 new documents not in the 2008 test ; *test* : 2008 test)

	50	50+	bc	test
Accuracy	80%	65.5%	26.9%	33%
MRR	0.84	0.68	0.31	0.42
Recall	90%	71.2%	38.7%	56%

TAB. 5 – **T4a task (English, EPPS data** : Results on the 3 different corpus and the test (*50* : 50 official development questions ; *50+* : 277 reformulated questions ; *bc* : Blind Corpus, 186 questions on the development documents ; *test* : 2008 test)

	50	50+	bc	test
Accuracy	68%	65.4%	36.1%	33%
MRR	0.76	0.71	0.45	0.36
Recall	88%	79.7%	61.1%	42%

TAB. 6 – **T5a task, Spanish, EPPS data** : Results on the 3 different corpus (*50* : 50 official development questions ; *50+* : 217 reformulated questions ; Blind Corpus, 36 questions, development document + one other document not in the 2008 test ; *test* : 2008 test

	Information Retrieval			Answer Extraction		
Task	Acc.	MRR	Recall	Acc.	MRR	Recall
T1a	43%	0.50	58%	41%	0.45	52%
T2a	46%	0.53	62%	33%	0.40	51%
T3a	69%	0.75	84%	45%	0.49	58%
T4a	53%	0.62	73%	33%	0.42	56%
T5a	50%	0.56	65%	33%	0.36	42%

TAB. 7 – Comparison between Information Retrieval module and answer extraction and scoring module

Table 7 gives the results for information retrieval and answer extraction and scoring allowing a direct comparison between them. A quick analysis of the problems have shown us that 3 main error sources were present :

– Poor quality of the answer scoring. Intrinsically, working only with distances and redundancy is not enough (especially with such a small number of documents as in QAst), dependencies in particular would probably be a big help.
– For T1 and T2, large differences between the development and test data, in particular related to the definition questions, made for over-specialisation in some of the routing rules and poor tuning.
– Some analysis errors, especially in Spanish and English, resulted in making some answers impossible to extract by the system. The analysis in better in French (T3) and it shows.

While the first and last point are entirely due to the system, the second one could have been avoided if the development data had been more representative of the test data.

4.2.2 General results on automatic transcripts

We did not do anything specific in order to handle recognition errors in the documents, the systems have been used as-is. As such our results show the loss due to the ASR on a decent but non-adapted system. The T3b, T4b and T5b tasks provided three different ASR outputs allowing an analysis of the impact of WER on the overall QA results. Table 8 gives the results on the ASR output depending on the task, the word error rate and the accuracy obtained on the respective manual transcriptions. The WERs for the T1b and T2b tasks are unknown.

	ASR_A		ASR_B		ASR_C		MAN
	Acc.	WER	Acc.	WER	Acc.	WER	Acc.
T3	41%	11%	25%	23.9%	21%	35.4%	45%
T4	21%	10.6%	20%	14%	19%	24.1%	33%
T5	24%	11.5%	19%	12.7%	23%	13.7%	33%

TAB. 8 – **Comparative results** for T3b, T4b, T5b and corresponding manual data ; *Acc.* : % correct answers in first rank ; *WER* : Word Error Rate

The better quality, including robustness, on the French analysis shows up immediatly again, the loss at equivalent error rate being roughly halved (5% instead of 10% at 11% WER). The loss rate does not seem to be easily predictable from the WER, but there are not enough data points to be sure. It may just be that 100 questions and a small number of documents is not enough to compute reliable statistics. A deeper analysis measuring the word error rate by word category could provide some intersting insights.

5 Conclusion

We presented the LIMSI question-answering systems on speech transcripts which participated to the QAst 2008 evaluation. These systems are based on a complete and multi-level language dependant analysis of both queries and documents followed by a language independant information retrieval and answer extraction and scoring. These systems obtained state-of-the-art results on the different tasks and languages.

Références

[1] Christelle Ayache, Brigitte Grau, and Anne Vilnat. Evaluation of question-answering systems : The French EQueR-EVALDA Evaluation Campaign. In *Proceedings of LREC'06*, Genoa - Italy, 24-26 May 2006.

[2] D.M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble : a high-performance learning name-finder. In *Proceedings of ANLP'97*, 1997.

[3] D. Déchelotte, H. Schwenk, G. Adda, and J.-L. Gauvain. Improved machine translation of speech-to-text outputs. Antwerp. Belgium, 2007.

[4] D. Giampiccolo, P. Forner, A. Peñas, C. Ayache, D. Cristea, V. Jijkoun, P. Osenova, P. Rocha, B. Sacaleanu, and R. Sutcliffe. Overview of the CLEF 2007 Multilingual Question Answering Track. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary, September 2007.

[5] H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of COLING*, 2002.

[6] M. Surdeanu, J. Turmo, and E. Comelles. Named entity recognition from spontaneous open-domain speech. In *in InterSpeech'05*, Lisbon, Portugal, 2005.

[7] E. M. Voorhees and L. P. Buckland. The Sixteenth Text REtrieval Conference Proceedings (TREC 2007). In Voorhees and Buckland, editor, *NIST Special Publication 500-274*, 2007.

[8] F. Wolinski, F. Vichot, and B. Dillet. Automatic processing of proper names in texts. In *Proceedings of EACL'95*, 1995.