The LIMSI participation to the QAst track

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

Abstract

In this paper, we present twe two different question-answering systems on speech transcripts which participated to the QAst 2007 evaluation. These two systems are based on a complete and multi-level analysis of both queries and documents. The first system uses handcrafted rules for small text fragments (snippet) selection and answer extraction. The second one replaces the handcrafting with 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. The evaluation results are ranged from 17% to 39% as accuracy depending on the tasks.

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

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, speech transcriptions of meeting and lectures

1 Introduction

In the QA and Information Retrieval domains progress has been demonstrated via evaluation campaigns for both open domain and limited domains [1, 2, 3]. In these evaluations systems are presented with independent 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 heavy syntactic and semantic analysis of both the question and the document or snippets given by search engine in which the answer has to be found. Such analysis can't reliably be performed on the data we are interested in. Typical textual QA systems are composed of question analysis, information retrieval and answer extraction components [1, 4]. The answer extraction component is quite complex and involves natural language analysis, pattern matching and sometimes even logical inference [5]. Most of these natural language tools are not designed to handle spoken phenomena.

In this paper, we present the architecture of the two QA systems developed in LIMSI for the QAst evaluation. Our QA systems are part of an interactive and bilingual (English and French) QA system called Ritel [6] which specifically addressed speed issues. The following sections present the documents and queries pre-processing and the non-contextual analysis which are common to both systems. The section 3 describes the older system (System 1). Section 4 presents the new system (System 2). Section 5 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 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 specificities 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. In order to use the same analysis module for all kinds of data, we should transform the query and the documents, which may come from different modality (text, manual transcripts, automatic transcripts) in order to have a common representation of the sentence, word, etc. This process is the 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, 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.

In the QAst evaluation, four data types are of interest:

- CHIL lectures [7] with manual transcriptions, where manual punctuations are separated from words. Only the splitting step is needed.
- CHIL lectures with automatic transcriptions [8]. Requires adding punctuation and splitting.
- AMI meetings [9] manual transcriptions. The transcriptions had been *textified*, with punctuation joined to the words, first words sentences upper-cased, etc. Requires all the steps except adding punctuation.
- AMI meetings with automatic transcriptions [10]. Lacking case, they required the last 3 steps.

Reconstructing the case and adding punctuation is done in the same process based on using a fully-cased, punctuated language model [11]. 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 Non contextual analysis module

The analysis is considered *non-contextual* because each sentence is processed in isolation. The general objective of this analysis is to find the bits of information that may be of use for search and extraction, which we call *pertinent information chunks*. These can be of different categories: named entities, linguistic entities (e.g. verbs, prepositions), or specific entities (e.g. scores). All words that do not fall into such chunks are automatically grouped into chunks via a longest-match strategy. Some examples of pertinent information chunks are given in Figure 1. In the following sections, the types of entities handled by the system are described, along with how they are recognized.

_prep in | _org NIST | _NN metadata evaluations | _verb reported | _NN speaker tracking
_score error rates | _aux are | _prep about | _val_score 15 %

Figure 1: Examples of pertinent information chunks from the CHIL data collection

2.2.1 Definition of Entities

Following commonly adopted definitions, the named entities are expressions that denote locations, people, companies, times, and monetary amounts. These entities have commonly known and accepted names. For example if the country France is a named entity, “capital of France” is not a named entity. However our experience is that the information present in the named entities is not sufficient to analyze the wide range of user utterances that can be found in lectures or meetings transcripts. Therefore we defined a set of specific entities in order to collect all observed information expressions contained in a corpus questions and texts from a variety of sources (proceedings, transcripts of lectures, dialogs etc.). Figure 2 summarizes the different entity types that are used.

Type of entities	Examples
classical named entities	`pers`: Romano Prodi ; Winston Churchill
	`prod`: Pulp Fiction ; Titanic
	`time`: third century ; 1998 ; June 30th
	`org`: European Commission ; NATO
	`loc`: Cambridge ; England
extended named entities	`method`: HMM, Gaussian mixture model
	`event`: the 9th conference on speech communication and technology
	`amount`: 500 ; two hundred and fifty thousand
	`measure`: year ; mile ; Hertz
	`color` red, spring green
question markers	`Qpers`: who wrote... ; who directed Titanic
	`Qloc`: where is IBM
	`Qmeasure`: what is the weight of the blue spoon headset
linguistic chunk	`compound`: language processing ; information technology
	`verb`: Roberto Martinez now **knows** the full size of the task
	`adj_comp`: the microphones would be **similar to** ...
	`adj_sup`: the **biggest** producer of cocoa of the world

Figure 2: Examples of the main entity types

2.2.2 Automatic detection of typed entities

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 learn-

ing techniques have been proposed [12, 13, 14]. 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. [15]) 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 and the processing time of most of existing linguistic analyzers is not compatible with the high speed we require.
We decided to tackle the problem with rules based on regular expressions on words as in other works [16]: we allow the use of lists for initial detection, and the definition of local contexts and simple categorizations. The tool used to implement the rule-based automatic annotation system is called `Wmatch`. 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). `Wmatch` includes also NLP-oriented features like strategies for prioritizing rule application, recursive substitution modes, word tagging (for tags like noun, verb...), word categories (number, acronym, proper name...). It has multiple input and output formats, including an XML-based one for interoperability and to allow chaining of instances of the tool with different rule sets. Rules are pre-analyzed and optimized in several ways, and stored in compact format in order to speed up the process. Analysis is multi-pass, and subsequent rule applications operate on the results of previous rule applications which can be enriched or modified. The full analysis comprises some 50 steps and takes roughly 4 ms on a typical user utterance (or document sentence). The analysis provides 96 different types of entities. Figure 3 shows an example of the analysis on a query (top) and on a transcription (bottom).

```
<_Qorg> which organization </_Qorg> <_action> provided </_action>
<_det> a </_det> <_NN> significant amount </_NN>
<_prep> of </_prep> <_NN> training data </_NN> <_punct> ? </_punct>
```

```
<_pro> it </_pro> <_verb> 's </_verb> <_adv> just </_adv>
<_prep_comp> sort of </_prep_comp> <_det> a </_det>
<_NN> very pale </_NN> <_color> blue </_color> <_conj> and </_conj>
<_det> a </_det> <_adj> light-up </_adj> <_color> yellow </_color>
<_punct> . </_punct>
```

Figure 3: Example annotation of a query: *which organization provided a significant amount of training data ?* (top) and of a transcription *it's just sort of a very pale blue* (bottom).

3 Question-Answering System 1

The *Question-Answering* system handles search in documents of any types (news articles, web documents, transcribed broadcast news, etc.). For speed reasons, the documents are all available locally and preprocessed: they are first normalized, and then analyzed with the NCA module. The (type, values) pairs are then managed by a specialized indexer for quick search and retrieval. This somewhat bag-of-typed-words system [6] works in three steps:

1. **Document query lists creation**. Using the entities found in the question, we generate a document query, and a ordered list of handcrafted back-off queries. These queries are obtained by relaxing some of the constraints on the presence of the entities, using a relative importance ordering (Named entity > NN > adj_comp > action > subs ...)

2. **Snippet retrieval:** we submit each query, according to their rank, to the indexation server, and stop as soon as we get document snippets (sentence or small groups of consecutive sentences) back.

3. **Answer extraction and selection:** the detection of the answer type has been extracted beforehand from the question, using Question Marker, Named, Non-specific and Extended Entities co-occurrences (_Qwho → _pers or _pers_def or _org). Therefore, we select the entities in the snippets with the expected type of the answer. At last, a clustering of the candidate answers is done, based on frequencies. The most frequent answer wins, and the distribution of the counts gives an idea of the confidence of the system in the answer.

4 Question-Answering System 2

System 1 has three main problems:

- The back-off queries lists require a large amount of maintenance work and will never cover all of the combinations of entities which may be found in the questions.

- The answer selection uses only frequencies of occurrence, often ending up with lists of first-rank candidate answers with the same score.

- The system answering speed directly depends on the number of snippets to retrieve which may sometimes be very large. To limit the number of snippets is not easy, as they are not ranked according to pertinence.

A new system, System 2 has been designed to solve these problems. We have kept the three steps described in section 3, with some major changes. In step 1, instead of instantiating document queries from a large number of preexisting handcrafted rules (about 5000), we generate a research descriptor using a very small set of rules (about 10); this descriptor contains all the needed information about the entities and the answer types, together with weights. In step 2, a score is calculated from the proximity between the research descriptor and the document and snippets, in order to choose the most relevant ones. In step 3, the answer is selected according to a score which takes into account many different features and tuning parameters, which allow an automatic and efficient adaptation.

4.1 Research Descriptor generation

The first step of System 2 is to build a research descriptor (data descriptor record, DDR) which contains the important elements of the question, and the possible answer types with associated weight. Some elements are marked as *critical*, which makes them mandatory in future steps, while others are *secondary*. The element extraction and weighting is based on a empirical classification of the element types in importance levels. Answer types are predicted through rules based on combinations of elements of the question. The Figure 4 shows an example of a DDR.

4.2 Documents and snippets selection and scoring

Each of the document is scored with geometric mean of the number of occurrences of all the DDR elements which appear in it. Using a geometric mean prevents from rescaling problems due to some elements being naturally more frequent. The documents are sorted by score and the n-best ones are kept. The speed of the entire system can be controlled by choosing n, the whole system being in practice `io`-bound rather than `cpu`-bound.
The selected documents are then loaded and all the lines in a predefined window (2-10 lines depending on question types) from the critical elements are kept, creating snippets. Each snippet is scored using the geometrical mean of the number of occurrences of all the DDR elements which appear in the snippet, smoothed with the document score.

```
{
question: in which company Bart works as a project manager ?
ddr:
{ w=1, critical, pers, Bart},
{ w=1, critical, NN, project manager },
{ w=1, secondary, action, works },
answer_type = {
  { w=1.0, type=orgof },
  { w=1.0, type=organisation },
  { w=0.3, type=loc },
  { w=0.1, type=acronym },
  { w=0.1, type=np },
}
```

Figure 4: Example of a DDR constructed from the question *in which company Bart works as a project manager*; each element contains a weight `w`, their importance for future steps, and the pair (type,value); each possible answer type contains a weight `w` and the type of the answer.

4.3 Answer extraction, scoring and clustering

In each snippet all the elements which type is one of the predicted possible answer types are candidate answers. We associate to each candidate answer A a score $S(A)$:

$$S(A) = \frac{[w(A)\sum_E \max_{e=E} \frac{w(E)}{(1+d(e,A))^\alpha}]^{1-\gamma} \times S_{snip}^\gamma}{C_d(A)^\beta C_s(A)^\delta} \tag{1}$$

In which:

- $d(e, A)$ is the distance to each element e of the snippet, instantiating a search element E of the DDR
- C_s is the number of occurrences of A in the extracted snippets, C_d in the whole document collection
- S_{snip} is the extracted snippet score (see 4.2)
- $w(A)$ is the weight of the answer type and $w(E)$ the weight of the element E in the DDR
- α, β, γ and δ are tuning parameters estimated by systematic trials on the development data. $\alpha, \beta, \gamma \in [0, 1]$ and $\delta \in [-1, 1]$

An intuitive explanation of the formula is that each element of the DDR adds to the score of the candidate ($\sum_E$) proportionally to its weight ($w(E)$) and inversely proportionally to its distance of the candidate($d(e, A)$). If multiple instance of the element are found in the snippet only the best one is kept ($\max_{e=E}$). The score is then smoothed with the snippet score (S_{snip}) and compensated in part with the candidate frequency in all the documents (C_d) and in the snippets (C_s).
The scores for identical (type,value) pairs are added together and give the final scoring for all the possible candidate answers.

5 Evaluation

In this section, we present the results obtained in the four tasks. T1 and T2 tasks were composed of an identical set of 98 questions; T3 task was composed of a different set of 96 questions and T4 task of a subset of 93 questions. Table 1 show the overall results with the 3 measures used in this evaluation. We submitted two runs, one for each system, for each of the four tasks. As required by the evaluation procedure, a maximum of 5 answers per question was provided.

Globally, we can see that System 2 gets better results than System 1. The improvement of the Recall (9-11%) observed on T1, and T3 tasks for System 2 illustrates that automatic generation

Task	System	Acc.	MRR	Recall
T1	Sys1	32.6%	0.37	43.8%
	Sys2	39.7%	0.46	57.1%
T2	Sys1	20.4%	0.23	28.5%
	Sys2	21.4%	0.24	28.5%
T3	Sys1	26.0%	0.28	32.2%
	Sys2	26.0%	0.31	41.6%
T4	Sys1	18.3%	0.19	22.6%
	Sys2	17.2%	0.19	22.6%

Table 1: General Results. *Sys1* System 1; *Sys2* System 2; *Acc.* is the accuracy, MRR is the Mean Reciprocal Rank and Recall the total number of correct answers in the 5 returned answers

of document/snippet queries greatly improves the coverage as compared to handcrafted rules. System 2 did not perform better than System 1 on the T2 task. Further analysis is needed to understand why.

The different modules we can evaluate are the analysis module, the passage retrieval and the answer extraction. The passage retrieval is easier to evaluate for System 2 because it is a complete separate module, which is not the case in the System 1. The Table 2 give the results on the passage retrieval in two conditions: with a limitation of the number of passages at 5 and without limitation. The diference between the Recall on the snippets (how often the answer is present in the selected snippets) and the QA Accuracy show that the extraction and the scoring of the answer has a reasonnable margin for improvement. The difference between the snippet Recall and its Accuracy (from 26 to 38% for the no limit condition) illustrates that the snippet scoring can be improved.

	Passage limit = 5			Passage without limit		
Task	Acc.	MRR	Recall	Acc.	MRR	Recall
T1	44.9%	0.52	67.3%	44.9%	0.53	71.4%
T2	29.6%	0.36	46.9%	29.6%	0.37	57.0%
T3	30.2%	0.37	47.9%	30.2%	0.38	68.8%
T4	18.3%	0.22	31.2%	18.3%	0.24	51.6%

Table 2: Results for Passage Retrieval for System 2. *Passage 5* the maximum of passage number is 5; *Passage without limit* there is no limit for the passage number; *Acc.* is the accuracy, MRR is the Mean Reciprocal Rank and Recall the total number of correct answers in the returned answers

One of the key uses of the analysis results is routing the question which is determining a rough class for the type of the answer (*language, location, ...*). The results of the routing component are given in Table 3 with details by answer category. Two questions of T1/T2 and three of T3/T4 were not routed.

We observed large differences with the results obtained on the development data, in particularly with the *method*, *color* and *time* categories. The analysis module has been built on corpus observations and it seems to be too dependant on the development data. That can explain the absence of major differences between System 1 and System 2 for the T1/T2 tasks. Most of the wrongly routed questions have been routed to the generic answer type class. In System 1 this class selects specific entities (*method, models, system, language...*) over the other entity types for the possible answers. In System 2 no such adaptation to the task has been done and all possible entity types have equal priority.

		All	LAN	LOC	MEA	MET	ORG	PER
T1/T2	% Correct	72%	100%	89%	75%	17%	95%	89%
	# Questions	98	4	9	28	18	20	9
T3/T4	% Correct	80%	100%	93%	83%	-	85%	80%
	# Questions	96	2	14	12	-	13	15

		TIM	SHA	COL	MAT
T1/T2	% Correct	80%	-	-	-
	# Questions	10	-	-	-
T3/T4	% Correct	71%	89%	73%	50%
	# Questions	14	9	11	6

Table 3: Routing evaluation. *All:* all questions; *LAN:* language; *LOC:* location; *MEA:* measure; *MET:* method/system; *ORG:* organization; *PER:* person; *TIM:* time; *SHAP:* shape; *COL*: colour.

6 Conclusion and future work

We presented the Question Answering systems used for our participation to the QAst evaluation. Two different systems have been used for this participation. The two main changes between System 1 and System 2 are the replacement of the large set of hand made rules by the automatic generation of a research descriptor, and the addition of an efficient scoring of the candidate answers. The results show that the System 2 outperforms the System 1. The main reasons are:

1. Better genericity through the use of a kind of expert system to generate the research descriptors.
2. More pertinent answer scoring using proximities which allows a smoothing of the results.
3. Presence of various tuning parameters which enable the adaption of the system to the various question and document types.

These systems have been evaluated on different data corresponding to different tasks. On the manually transcribed lectures, the best result is 39% for Accuracy, on manually transcribed meetings, 24% for Accuracy. There was no specific effort done on the automatically transcribed lectures and meetings, so the performances only give an idea of what can be done without trying to handle speech recognition errors. The best result is 18.3% on meeting and 21.3% on lectures. From the analysis presented in the previous section, performance can be improved at every step. For example, the analysis and routing component can be improved in order to better take into account some type of questions which should improve the answer typing and extraction. The scoring of the snippets and the candidate answers can also be improved. In particular some tuning parameters (like the weight of the transformations generated in the DDR) have not been optimized yet.

7 Acknowledgments

This work was partially funded by the European Commission under the FP6 Integrated Project IP 506909 Chil and the LIMSI AI/ASP Ritel grant.

References

[1] E. M. Voorhees, L. P. Buckland. The Fifteenth Text REtrieval Conference Proceedings (TREC 2006), In Voorhees and Buckland eds. 2006.

[2] B. Magnini, D. Giampiccolo, P. Former, C. Ayache, P. Osenova, A. Penas, V. Jijkown, B. Sacaleanu, P. Rocha, R. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. Working Notes for the CLEF 2006 Workshop. 2006.

[3] C. Ayache, B. Grau, A. Vilnat. Evaluation of question-answering systems : The French EQueR-EVALDA Evaluation Campaign. Proceedings of LREC'06, Genoa, Italy.

[4] S. Harabagiu and D. Moldovan. Question-Answering. In *The Oxford Handbook of Computational Linguistics*. R. Mitkov (Eds). Oxford University Press. 2003.

[5] S. Harabagiu, A. Hickl. Methods for using textual entailment in Open-Domain question-answering. Proceedings of COLING'06. Sydney, Australia. July 2006.

[6] B. van Schooten, S. Rosset, O. Galibert, A. Max, R. op den Akker, G. Illouz. Handling speech input in the Ritel QA dialogue system. 2007. Proceedings of Interspeech'07. Antwerp. Belgium. August 2007.

[7] CHIL Project. http://chil.server.de

[8] L. Lamel, G. Adda, E. Bilinski, and J.-L. Gauvain. Transcribing Lectures and Seminars. In InterSpeech, Lisbon, September 2005.

[9] AMI project. http://www.amiproject.org

[10] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan. The AMI Meeting Transcription System: Progress and Performance. Rich Transcription 2006 Spring (RT06s) Meeting Recognition Evaluation. 3 May 2006, Bethesda, Maryland, USA.

[11] D. Déchelotte, H. Schwenk, G. Adda, J.-L. Gauvain. Improved Machine Translation of Speech-to-Text outputs. 2007. Proceedings of Interspeech'07. Antwerp. Belgium. August 2007.

[12] D.M. Bikel, S. Miller, R. Schwartz, R. Weischedel. Nymble: a high-performance learning name-finder. Proceedings of ANLP'97, Washington, USA, 1997.

[13] H. Isozaki, H. Kazawa, Efficient Support Vector Classifiers for Named Entity Recognition. Proceedings of COLING, Taipei. 2002.

[14] M. Surdeanu, J. Turmo, E. Comelles. Named Entity Recognition from spontaneous Open-Domain Speech. Proceedings of InterSpeech'05, Lisbon, Portugal. 2005.

[15] F. Wolinski, F. Vichot, B. Dillet. Automatic Processing of Proper Names in Texts. Proceedings of EACL'95, Dublin, Ireland. 1995.

[16] S. Sekine. Definition, dictionaries and tagger of Extended Named Entity hierarchy. Proceedings of LREC'04, Lisbon, Portugal. 2004.