THE LIMSI QAST SYSTEMS: COMPARISON BETWEEN HUMAN AND AUTOMATIC RULES GENERATION FOR QUESTION-ANSWERING ON SPEECH TRANSCRIPTIONS

Sophie Rosset, Olivier Galibert, Gilles Adda, Eric Bilinski

Spoken Language Processing Group, LIMSI-CNRS, B.P. 133, 91403 Orsay cedex, France
{rosset, galibert, gadda, bilinski}@limsi.fr

ABSTRACT

In this paper, we present two different question-answering systems on speech transcripts. 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 preliminary results obtained on QAst (QA on speech transcripts) development data are promising ranged from 72% correct answer at 1st rank on manually transcribed meeting data to 94% on manually transcribed lecture data.

Index Terms— Question answering, speech recognition of meetings and lectures

1. INTRODUCTION

Searching for information can be done using one of two main paradigms: document retrieval and information extraction. In the first approach, documents matching a user query, (most often the match is done on some keywords extracted from the query), are returned. Based on the assumption that the theme of these documents is the one that is best described by the query, they constitute a pool in which the user might find information that may meet some need. This need can be very specific (e.g. Who is presiding the French Senate?), or it can be theme-oriented (e.g. I’d like information about the French Senate). The second approach to search is embodied by so-called question answering systems (QA), which return the most probable answer given a specific spelled out question (e.g. Who won the 2005 Tour de France? Lance Armstrong.). In the QA and Information Retrieval domains progress has been observed via evaluation campaigns [1, 2, 3]. In these evaluations, the systems handle independent questions and should provide one answer to each question, extracted from textual data, for both open domain and limited domain. Therefore, as a large part of human interactions happens through speech, e.g. meetings, seminars, lectures, telephone conversations, current factual QA systems need a deep adaptation to be able to access the information contained in these data.

Spoken data is different from textual data in various ways: it contains disfluencies, false starts, speaker corrections, truncated words. The grammatical structure of spontaneous speech is quite different than for written discourse. Moreover, the data we want to process are meetings which show a complete different global structure (for instance, interaction creates run-on sentences where the distance between the first part of an utterance and the last one can be very long). Most of the QA systems use a complete and heavy syntactic and semantic analysis of both the question and the document or text fragments given by search engine (snippet) 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. Recently, within the CHIL project, some studies have been done in the field of question-answering on spoken data [6]. As a follow-up of that study, a new pilot track called QAst (Question-Answering on Speech Transcripts) has been organized as part of the CLEF evaluation [7].

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 a bilingual (English and French) Interactive QA system called Ritel [8] and as such the speed aspect has specifically been taken into account. The next section presents the QAst evaluation and data. The following sections present the documents and queries pre-processing and the non-contextual analysis which are common to both systems. The section 4 describes the older system (System 1). Section 5 presents the new system (System 2). Section 6 finally presents some preliminary results for these two systems.

2. QAST EVALUATION AND DATA

The objective of this evaluation is to provide a framework in which QA systems can be evaluated when the answers have
to be found in spontaneous speech transcriptions (manual and automatic transcriptions). Four tasks have been defined:

- **T1**: QA in manual transcriptions of lectures.
- **T2**: QA in automatic transcriptions of lectures.
- **T3**: QA in manual transcriptions of meetings.
- **T4**: QA in automatic transcriptions of meetings.

The possible answers of the questions can be of different types: person, location, organization, language, system, method, measure, time, color, shape, material. The development data contains: (a) 10 lectures from CHIL [9], containing 61,000 words, and 50 questions (domain of the lectures: speech and language processing) and (b) 50 meetings from AMI [10], containing 307,347 words, and 50 questions (domain of the meetings: design of television remote control). 10% of every question set has no answer in the documents. The automatic transcripts of the CHIL lectures has been provided by the LIMSI [11].

3.1. 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 could be:

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 with manual transcriptions, where manual punctuations are separated from words. Only the splitting step is needed.
- CHIL lectures with automatic transcriptions. Requires adding punctuation and splitting.
- AMI meetings 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. 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 [12]. 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, was then passed to the non-contextual analysis.

3.2. 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 also annotated by following a longest-match strategy to find chunks with coherent meanings. 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.

3.2.1. Definition of Entities

Following commonly adopted definitions, the named entities are expressions that denote locations, people, companies, ti-
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 have been proposed [13, 14, 15]: 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. [16]) 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 [17]: 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 and Figure 4 on a transcription.

![Fig. 2. Examples of the main entity types](image)

![Fig. 3. Annotation of a query: which organization provided a significant amount of training data?](image)

![Fig. 4. Annotation of a transcription: it’s just sort of a very pale blue and a light-up yellow.](image)

### 4. 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 index for quick search and retrieval.
This somewhat bag-of-typed-words system [8] 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 types 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.

5. **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 4, 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.

### 5.1. 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 an 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 5 shows an example of a DDR.

**Fig. 5.** Example of a DDR constructed from the question *in which company Bart works as a project manager?*

```plaintext
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=0.1, type=loc },
{ w=0.1, type=acronym },
{ w=0.1, type=acronym },
}
```

5.2. **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 1-o-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.

5.3. **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):

$$S(A) = \frac{\sum_{E} \max_{e=E} \left[ \frac{w(E)}{1+d(e,A)^\alpha} \right]^{1-\gamma} \times S^\gamma_{snip}}{C_d(A)^3 C_s(A)^8}$$

In which:

- $d(e, A)$ is the distance to each element $e$ of the snippet,
instantiating a search element $E$ of the DDR

- $C_A$ 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 5.2)
- $w(A)$ is the weight of the answer type and $w(E)$ the weight of the element $E$ in the DDR

- $\alpha, \beta, \gamma$ and $\delta$ 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 ($\max_{e \in E}$). 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 ($C_A$).

The scores for identical (type,value) pairs are added together and give the final scoring for all the possible candidate answers.

### 6. PRELIMINARY EVALUATION

Both systems are being tested in the 2007 QAst evaluation. We present here the results on the development data of the QAst evaluation. The QA systems are evaluated on four different tasks: QA on manually transcribed lectures (T1), QA on automatically transcribed lectures (T2), QA on manually transcribed meetings (T3) and QA on automatically transcribed meetings (T4). Table 1 shows the results of System 1 and System 2 on the CHIL data (lectures, manual and automatic transcriptions). Table 2 shows the results of System 1 and System 2 on the AMI data (meetings, manual and automatic transcriptions). The metrics used for this evaluation are: the percentage of correct answers given in the first position (1st Rank), the mean reciprocal rank on the 5 first answers (MRR). In order to examine the loss caused by the answer extraction and scoring, we added the percentage of correct answers regardless of their ranking (Recall). This last measure gives a good idea of what the results could be if the answer scoring was perfect.

<table>
<thead>
<tr>
<th>Task</th>
<th>System</th>
<th>Cond.</th>
<th>1st Rank</th>
<th>MRR</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Sys1</td>
<td>nil</td>
<td>56%</td>
<td>0.58</td>
<td>60%</td>
</tr>
<tr>
<td>T1</td>
<td>Sys1</td>
<td>norm</td>
<td>74%</td>
<td>0.79</td>
<td>84%</td>
</tr>
<tr>
<td>T1</td>
<td>Sys1</td>
<td></td>
<td>32.6%</td>
<td>0.37</td>
<td>43.8%</td>
</tr>
<tr>
<td>T1</td>
<td>Sys2</td>
<td>nil</td>
<td>66%</td>
<td>0.70</td>
<td>76%</td>
</tr>
<tr>
<td>T1</td>
<td>Sys2</td>
<td>norm</td>
<td>94%</td>
<td>0.95</td>
<td>98%</td>
</tr>
<tr>
<td>T1</td>
<td>Sys2</td>
<td></td>
<td>39.7%</td>
<td>0.46</td>
<td>57.1%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys1</td>
<td>nil</td>
<td>32%</td>
<td>0.38</td>
<td>42%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys1</td>
<td>norm</td>
<td>24%</td>
<td>0.30</td>
<td>36%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys1</td>
<td></td>
<td>20.4%</td>
<td>0.23</td>
<td>28.5%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys2</td>
<td>nil</td>
<td>32%</td>
<td>0.34</td>
<td>38%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys2</td>
<td>norm</td>
<td>34%</td>
<td>0.35</td>
<td>36%</td>
</tr>
<tr>
<td>T2</td>
<td>Sys2</td>
<td></td>
<td>21.4%</td>
<td>0.24</td>
<td>28.5%</td>
</tr>
</tbody>
</table>

Table 1. Results on CHIL (seminar) data. Sys1 System 1; Sys2 System 2; norm. with data normalization; T1: manual transcripts; T2: ASR transcripts

<table>
<thead>
<tr>
<th>Task</th>
<th>System</th>
<th>Cond.</th>
<th>1st Rank</th>
<th>MRR</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3</td>
<td>Sys1</td>
<td>nil</td>
<td>18%</td>
<td>0.22</td>
<td>26%</td>
</tr>
<tr>
<td>T3</td>
<td>Sys1</td>
<td>norm</td>
<td>28%</td>
<td>0.36</td>
<td>48%</td>
</tr>
<tr>
<td>T3</td>
<td>Sys1</td>
<td></td>
<td>26.0%</td>
<td>0.28</td>
<td>32.2%</td>
</tr>
<tr>
<td>T3</td>
<td>Sys2</td>
<td>nil</td>
<td>34%</td>
<td>0.36</td>
<td>40%</td>
</tr>
<tr>
<td>T3</td>
<td>Sys2</td>
<td>norm</td>
<td>72%</td>
<td>0.76</td>
<td>84%</td>
</tr>
<tr>
<td>T3</td>
<td>Sys2</td>
<td></td>
<td>26.0%</td>
<td>0.31</td>
<td>41.6%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys1</td>
<td>nil</td>
<td>10%</td>
<td>0.10</td>
<td>10%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys1</td>
<td>min</td>
<td>18%</td>
<td>0.20</td>
<td>22%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys1</td>
<td>ci</td>
<td>18%</td>
<td>0.20</td>
<td>24%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys1</td>
<td>norm</td>
<td>20%</td>
<td>0.22</td>
<td>26%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys1</td>
<td></td>
<td>18.3%</td>
<td>0.19</td>
<td>22.6%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys2</td>
<td>nil</td>
<td>10%</td>
<td>0.10</td>
<td>10%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys2</td>
<td>min</td>
<td>18%</td>
<td>0.20</td>
<td>24%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys2</td>
<td>ci</td>
<td>14%</td>
<td>0.18</td>
<td>26%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys2</td>
<td>norm</td>
<td>32%</td>
<td>0.35</td>
<td>38%</td>
</tr>
<tr>
<td>T4</td>
<td>Sys2</td>
<td></td>
<td>17.2%</td>
<td>0.19</td>
<td>22.6%</td>
</tr>
</tbody>
</table>

Table 2. Results on AMI (meeting) data. Sys1 System 1; Sys2 System 2; norm. with data normalization; ci. with case independent analysis; min. with down-cased data; T3: manual transcripts; T4: ASR transcripts

could be given for this result:

- LIMSI ASR output which was provided as data for the T2 task is very close to the kind of data our analysis expects (i.e. the normalization process is not necessary).
- The re-punctuation process of the normalization tends to produce shorter segments than what the ASR gave which increases the distances for System 1 which uses much smaller snippets than System 2 because of the lack of scoring.

These results show that System 2 systematically outperforms System 1. The difference between System 2 and System 1 is larger on manually transcribed data, which shows that the potentiality of System 2 is weakened by the errors in the speech transcripts. On the CHIL data (T1 and T2 tasks) and for System 2, we may observe that the Recall is about equal to the 1st
rank, while System 1 exhibits an absolute difference of 10% between 1st rank and Recall; this result shows the efficiency of the new answer extraction and scoring, even if a further improvement could be obtained by a more efficient clustering. The improvement of the Recall (12-36%) observed on T1, T3 and T4 task for System 2 illustrates that automatic generation of document/snippet queries greatly improves the coverage as compared to handcrafted rules.

We observed large differences between development and test results (cf Tables 1 and 2), particularly with the method, color and time categories. One explanation is that the analysis module, developed on corpus observations, seems too dependent on the development data. 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.

7. CONCLUSION AND PERSPECTIVES

We presented two different systems and their results on the development data provided by the QASt evaluation campaign. The main two changes between System 1 and System 2 are the substitution of the large set of hand made rules by automatic generation of a research descriptor, and the adjunction of an efficient scoring of the candidate answers; both modifications lead to improved results. The results show clearly that the System 2 which we will call “Strategies QA system” systematically outperforms the System 1. The main reasons are:

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

These systems have been evaluated on different data corresponding to different tasks. On the manually transcribed lectures, the best result is 94% at the 1st Rank, on manually transcribed meetings, 72% at the 1st Rank. 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 a 32% on meeting and 34% on lectures.

While we observed a large discrepancy between results obtained on test and development data, we observed that System 2 still outperforms the System 1, and obtains very good results during the official evaluation.

8. ACKNOWLEDGMENTS

This work was partially funded by the European Commission under the FP6 Integrated Project IP 506909 CHIL.

9. REFERENCES