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 needs 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

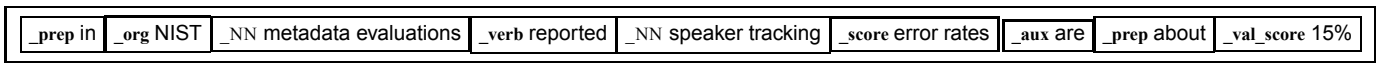

Fig. 1. Examples of pertinent information chunks from the CHIL data collection

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 transcripts 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]. The results presented in this paper concern only the development data.

3. 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.

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-

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

Fig. 2. Examples of the main entity types

mes, 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.

3.2.2. 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 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.

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

Fig. 3. Annotation of a query: *which organization provided a significant amount of training data ?*

<_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>

Fig. 4. Annotation of a transcription: *it's just sort of a very pale blue and a light-up yellow.*

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 indexer 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 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.

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 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 5 shows an example of a DDR.

```
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 },
}
```

Fig. 5. 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.

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 `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.

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 $S(A)$:

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

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 5.2)

• $w(A)$ 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.

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.

Tables 1 and 2 show that, for both meeting and lectures manually transcribed and both approaches, the normalization (see 3.1) allows a significant improvement of the results (from 10% to 32% depending on the task and the system). The T4 data was all uppercase. In order to correctly test our system without the normalization process, we performed the analysis on down-cased data and with a case-insensitive analysis system. Both conditions offered better results as the one with raw data, which gave correct answers only for the 10% NIL answers. The normalization process performed significantly better than those simple approaches and allow an interesting improvement. For the T2 task (QA on ASR transcripts of lectures), the normalization process degrades the results for System 1, and doesn't improve for System 2. Two reasons could be given for this result:

Task	System	Cond.	1st Rank	MRR	Recall
T1	Sys1	nil	56%	0.58	60%
T1	Sys1	norm	74%	0.79	84%
T1test	Sys1		32.6%	0.37	43.8%
T1	Sys2	nil	66%	0.70	76%
T1	Sys2	norm	94%	0.95	98%
T1test	Sys2		39.7%	0.46	57.1%
T2	Sys1	nil	32%	0.38	42%
T2	Sys1	norm	24%	0.30	36%
T2test	Sys1		20.4%	0.23	28.5%
T2	Sys2	nil	32%	0.34	38%
T2	Sys2	norm	34%	0.35	36%
T2test	Sys2		21.4%	0.24	28.5%

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

Task	System	Cond.	1st Rank	MRR	Recall
T3	Sys1	nil	18%	0.22	26%
T3	Sys1	norm	28%	0.36	48%
T3test	Sys1		26.0%	0.28	32.2%
T3	Sys2	nil	34%	0.36	40%
T3	Sys2	norm	72%	0.76	84%
T3test	Sys2		26.0%	0.31	41.6%
T4	Sys1	nil	10%	0.10	10%
T4	Sys1	min	18%	0.20	22%
T4	Sys1	ci	18%	0.20	24%
T4	Sys1	norm	20%	0.22	26%
T4test	Sys1		18.3%	0.19	22.6%
T4	Sys2	nil	10%	0.10	10%
T4	Sys2	min	18%	0.20	24%
T4	Sys2	ci	14%	0.18	26%
T4	Sys2	norm	32%	0.35	38%
T4test	Sys2		17.2%	0.19	22.6%

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

• 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 dependant 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

[1] Voorhees, E.M. TheFourteenth Text REtrieval Conference Proceedings (TREC 2005), In Voorhees and Buckland eds. 2005.

[2] A. Vallin, D. Giampiccolo, L. Aunimo, C. Ayache, P. Osenova, A. Pes, M. de Rijke,B. Sacaleanu, D. Santos, R. Sutcliffe. Overview of the CLEF 2005 Multilingual Question Answering Track. Working Notes for the CLEF 2005 Workshop, Vienna, Austria. 2005.

[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] M. Surdeanu, D. Dominguez-Sal, P.R. Comas. Design and performance analysis of a factoid question-answering system for spontaneous speech transcriptions. Proceedings of Interspeech'06. Pittsburgh. USA. September 2006.

[7] QAst on Speech Transcripts. CLEF 2007. http://http://www.lsi.upc.edu/∼qast/

[8] 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.

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

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

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

[12] 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.

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

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

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

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

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