Robust Question Answering for Speech Transcripts Using Minimal Syntactic Analysis

Pere R. Comas, Jordi Turmo and Mihai Surdeanu
TALP Research Center
Technical University of Catalonia (UPC)
{pcomas,turmo,surdeanu}@lsi.upc.edu

Abstract

This paper describes the participation of the Technical University of Catalonia in the CLEF 2007 Question Answering on Speech Transcripts track. For the processing of manual transcripts we have deployed a robust factual Question Answering that uses minimal syntactic information. For the handling of automatic transcripts we combine the QA system with a novel Passage Retrieval and Answer Extraction engine, which is based on a sequence alignment algorithm that searches for "sounds like" sequences in the document collection. We have also enriched the NERC with phonetic features to facilitate the recognition of named entities even when they are incorrectly transcribed.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]

General Terms

Experimentation, Performance, Measurement

Keywords

Question Answering, Spoken Document Retrieval, Phonetic Distance

1 Introduction

The CLEF 2007 Question Answering on Speech Transcripts (QAST) track consists of the following four tasks:

T1 : Question Answering (QA) using as underlying document collection the manual transcripts of the lectures recorded within the CHIL European Union project[1].

T2 : QA using the automatic transcripts of the CHIL lectures. Word lattices from an automated speech recognizer (ASR) are provided as an additional input source for systems that prefer to decide internally what the best automatic segmentation is.

T3 : QA in the manual transcriptions of the meetings that form the corpus collected by the AMI European Union project[2].

T4 : QA in the automatic transcripts of the above AMI meetings.

[1] http://chil.server.de

[2] http://www.amiproject.org

For tasks T1 and T3 we have adapted a QA system and Named Entity Recognizer and Classifier (NERC) that we previously developed for the processing of manual speech transcripts[9, 10]. Both these systems obtained good performance in previous evaluations even though they require minimal syntactic analysis of the underlying documents (only part of speech tagging) and minimal additional annotation (punctuation signs are optional). For the handling of automatic transcripts (tasks T2 and T4) we implemented two significant system changes: (a) for Passage Retrieval (PR) and Answer Extraction (AE) we designed a novel keyword matching engine that relies on phonetical similarity –instead of string match– to overcome the errors introduced by the ASR, and (b) we enriched the NERC with phonetic features to facilitate the recognition of named entities even when they are incorrectly transcribed. Even though the resulting QA system does not outperform the initial QA system in tasks T2 and T4, we believe these design choices are a good longer-term research direction because they can address ASR-specific phenomena.

The paper is organized as follows. Section 2 overviews the architecture of the QA system. Section 3 describes the NERC improvements, for both manual and automatic transcripts. Section 4 details the novel keyword matching algorithm we designed for automatic transcripts. Section 5 contains the results of the empirical evaluation and Section 6 concludes the paper.

2 Overview of the System Architecture

The architecture of our QA system follows a commonly-used schema, which splits the process into three phases that are performed sequentially: Question Processing (QP), Passage Retrieval (PR), and Answer Extraction (AE). In the next sub-section we describe the implementation of the three components for the system that processes manual transcripts. We conclude this section with the changes required for the handling of automatic transcripts.

2.1 QA System for Manual Transcripts

For the processing of manual transcripts we used an improved version of the system introduced in [9]. We describe it briefly below.

Question Processing. The main goal of this component is to detect the type of the expected answer (e.g., the name of a location, organization etc.). We currently recognize the 53 open-domain answer types from [7] and an additional 3 types that are specific to the corpora used in this evaluation (i.e., `system/method`, `shape`, and `material`). The answer types are extracted using a multi-class Perceptron classifier and a rich set of lexical, semantic (i.e., distributional similarity) and syntactic (part of speech (POS) tags and syntactic chunks) features. This classifier obtains an accuracy of 88.5% on the corpus of [7]. Additionally, the QP component extracts and ranks relevant keywords from the question (e.g., a noun is ranked as more important than a verb, stop words are skipped). Since questions are typed text in all QAST scenarios, we used the same QP component for both manual and automatic transcripts.

Passage Retrieval. The goal of this component is to retrieve a set of relevant passages from the document collection, given the previously extracted question keywords. The PR algorithm uses a query relaxation procedure that iteratively adjusts the number of keywords used for retrieval and their proximity until the quality of the recovered information is satisfactory (see [9]). In each iteration a Document Retrieval application[3] fetches the documents relevant for the current query and a subsequent passage construction module builds passages as segments where two consecutive keyword occurrences are separated by at most t words. Figure 1 shows an example of passage construction for a simple query and one sample sentence. This algorithm uses limited syntax –only POS tags– which makes it very robust for speech transcripts.

[3] Lucene - http://jakarta.apache.org/lucene

Keywords: relevant, documents, process

$$\textbf{\textit{documents}}\ \underbrace{\textit{must be separated into}}_{\text{distance}>t}\ \overbrace{\textbf{\textit{relevant documents}}\ \textit{and irrelevant}\ \textbf{\textit{documents}}\ \underbrace{\textit{by manual}}_{\text{distance}<t}\ \textbf{\textit{process,}}}^{\text{Passage}}\ \textit{which}\ldots$$

Figure 1: Example of passage building.

1M: *"The pattern frequency relevance rate indicates the ratio of relevant documents..."*
1A: *"the putt and frequency illustrating the case the ratio of relevant documents..."*
2M: *"Documents must be separated into relevant documents and irrelevant documents by a manual process, which is very time consuming."*
2A: *"documents must be separated into relevant documents and in relevant document by a manual process witches' of very time consuming"*
3M: *"The host system it is a UNIX Sun workstation"*
3A: *"that of system it is a unique set some workstation"*

Figure 2: Examples of manual (M) and automatic (A) transcripts.

Answer Extraction. This component identifies the exact answer to the given question within the passages retrieved by the previous module. First, answer candidates are identified as the set of named entities that occur in these passages and have the same type as the answer type detected by QP. Then, these candidates are ranked using a scoring function based on a set of heuristics that measure keyword distance and density[8].

2.2 QA System for Automatic Transcripts

The state of the art in ASR technology is far from perfect, especially when processing spontaneous speech. For example, the word error rate (WER) of the AMI automatic transcripts is around 38% and the WER of the CHIL transcripts is over 20%. Figure 2 shows several examples of common errors when generating automatic transcripts. From the point of view of QA, imperfect transcripts create the following problems:

- The keywords identified as relevant by QP define the context where the correct answer appears. Hence they are useful for the extraction of relevant documents and passages, and for the ranking of candidate answers. When these specific keywords are incorrectly transcribed by the ASR, all these tasks are in jeopardy.
- Most named entities that yield candidate answers appear as proper nouns with low frequency in the corpora. Due to this low frequency it is unlikely that the ASR language models include them (they will be marked as out of vocabulary (OOV) words). This increases the probability that the ASR incorrectly recognize the named entities relevant for the AE component.

In order to address these issues specific to automatically-generated transcripts we have developed a novel QA system by changing the PR, AE and NERC components. The main difference between the new PR and AE modules and those used to process manual transcripts is the strategy for keyword searching. Our hypothesis is that an approximated matching between the automatic transcripts and the question keywords, all of them phonetically transcribed, can perform better than classical IR techniques for written text. Under this assumption, the automatic transcripts of all corpus documents and the relevant question keywords extracted by QP are deterministically transformed to phonetic sequences. Then we use a novel retrieval engine named PHAST, which computes document (or passage or answer context) relevance based on approximated matching of phonetic sequences. While PHAST was initially developed for document retrieval, in the end we used the same algorithm to rank passages in PR and answer contexts in AE. PHAST is detailed in Section 4.

3 Named Entity Recognition and Classification

As described before, we extract candidate answers from the named entities (NEs) that occur in the passages retrieved by the PR component. We detail below the strategies used for NERC in both manual and automatic transcripts.

3.1 NERC for Manual Transcripts

Our initial idea for the identification of NEs in manual transcripts was to use the NERC we developed previously for the processing of speech transcripts [10]. One change from the previous system is that, for faster training times, we replaced the existing SVM classifiers with a multi-class Perceptron.[4] To verify the validity of this approach we annotated the NEs that occur in the QAST development corpus with their types (i.e., person, organization, location, language, measure, system/method and time) and used an 80–20% corpus split for training and testing for both CHIL and AMI corpora. This experiment indicated that the development data is sufficient for good generalization for AMI –we obtained a F_1 score of +75 points in the development test partition– but it is insufficient in CHIL: the model learned had a F_1 score below 33 points. This is most likely caused by the small size of the CHIL development corpus and the large number of topics addressed. To compensate for the insufficient CHIL training data we decided to perform a combination of several NERC models for this task. We merged the outputs of: (a) a rule-based NERC developed previously [9], (b) the above NERC trained on the existing CHIL development data, and (c) the above NERC trained on the CoNLL English corpus[5]. We used the above priority ordering for conflict resolution in case of overlapping assignments (e.g., the CHIL model has higher priority than the CoNLL model). After model combination the NERC F_1 score in the development test partition did not improve but the recall did increase, so we decided to use this combination strategy in the formal testing. We favored a NERC with higher recall in the detriment of precision because for the QA problem the NERC job is only identification of candidate answers, so recall is paramount.

3.2 NERC for Automatic Transcripts

We used a similar framework for the processing of automatic transcripts: we annotated the development corpora and trained specific NERC models for CHIL and AMI. The significant difference from the previous approach is that here we expand the classifiers' feature sets with phonetic attributes. These features are motivated by the fact that even when the ASR incorrectly transcribes NEs the phonetic structure is by and large maintained in the transcript. For example, in Figure 2 the organization name *"Sun"* is recognized as *"some"*, a token with almost the same phonetical structure. In this work we model the similarities between phonetic sequences as features. We used an unsupervised hierarchical clustering algorithm that groups together tokens based on the similarity of their phonetic sequences. The stop condition of the hierarchical clustering algorithm is selected to reach a local maximum of the Calinski criterion [2]. The cluster id of each token is then added as a feature in the NERC model. For example, *"Sun"* and *"some"* share the same cluster id, which helps the NERC model generalize from the correct to the incorrect transcript. We added phonetic features that model not only the complete words, but also their prefixes and suffixes.

4 The Phonetic Sequence Alignment Algorithm

This section describes PHAST, the phonetic sequence alignment algorithm we used for keyword matching. While here we illustrate PHAST in the context of document retrieval, we used the same algorithm for passage retrieval and identification of answer contexts. PHAST is based on

[4] The software is available for download here: http://bios-tagger.sourceforge.net

[5] http://cnts.ua.ac.be/conll2002/ner

Reference transcript: *"The host system it is a UNIX Sun workstation"*
Automatic transcript: *"that of system it is a unique set some workstation"*

	junik	← *detection*	
…ðæt ʌβ sɪstəm ɪt ɪz ə	junik	sɛt sʌm	wəʊrksteɪʃən…
	junik	s sʌn	← *extension*

Figure 3: Search of term "UNIX-Sun".

BLAST[1], an algorithm from the field of pattern matching in bio informatics, which we adapted to work with phone sequences instead of protein sequences. In our case, the input data is a transcript collection D transformed to phonetic sequences and a set of query terms KW also mapped to phonetic sequences.

Algorithm 1

PHAST algorithm

Parameter: $\mathcal{D}$, collection of phonetically transcribed documents
Parameter: $\mathcal{KW}$, set of phonetically transcribed keywords

```
for all $d \in \mathcal{D}, w \in \mathcal{KW}$ do
  while $h = detection(w, d)$ do
    $s = extension(w, h)$
    if $relevant(s, h)$ then
      mark $w$ as matched → update $tf(w, d)$
    end if
  end while
end for
```

PHAST is detailed in Algorithm 1. The procedure works as follows: function *detection*() detects subsequences of transcript d at phone number r with moderate resemblance with keyword w (a *weak similarity*), then *extension*() computes a similarity score s between d and w at r, and *relevant*() judges how this occurrence at r is relevant to term frequency. For detecting weak similarities function *detection*() uses a deterministic finite automaton (DFA) [3] to recognise substrings of fixed length n from w while scanning d. One DFA is used for each query word therefore all the keywords are searched in one pass. Our hypothesis is that in the automatic transcript the transcribed words will keep a phonetic resemblance with the original words and short sequences of n phones will be in the original position. Function *extension*() is a measure of phonetic similarity (see [4, 5]). We compute the similarity s of two sequences using the edit distance (Levenshtein distance [6]) with a cost function that measures inter-phone similarity. The score s is a bounded non-integer value that can be normalised into the interval $[0, 1]$ (i.e., for two identical sequences $s = 1.0$). Function *relevant*() considers a hit any matching with the score above some fixed threshold. In the context of document retrieval, term frequency is computed by adding the scores of these hits. For PR and AE we used all relevant matchings in the algorithms described in Section 2.1.

Figure 3 shows an example of how functions *detection* and *extension* are used. Document d is the sentence 3A from Figure 2, which has been transcribed to a sequence of phones. The query word w is the term *"UNIX-Sun"*, which is transcribed as [juniks sʌn][6]. Term w exists in the manual transcript 3M but not in the automatic transcript 3A. In the first step, *detection* finds hook [junik] related to [juniks sʌn]. In the second step, *extension* extends the hook by matching the rest of [juniks sʌn] with the phones surrounding [junik] in the automatic transcript.

[6] We use the international phonetic alphabet (IPA) within brackets: http://www.arts.gla.ac.uk/IPA/

Table 1: Overall results for the four QAST tasks. For task T3 we report scores using a post-deadline submission where some bugs in our output formatting script were fixed.

Task and System	MRR	TOP1
T1, QA$_m$	0.53	0.51
T2, QA$_a$	0.25	0.24
T2, QA$_m$	0.37	0.36
T3, QA$_m$	0.26	0.25
T4, QA$_a$	0.15	0.13
T4, QA$_m$	0.22	0.21

5 Experimental Results

UPC participated in all four tasks organized under QAST. Initially, each QAST task included 100 test questions, but a few questions were removed in the final evaluation due to various problems, e.g., answer types outside of the accepted set, repetitions, etc. The final question distribution was: 98 questions in T1 and T2, 96 in T3, and 93 in T4. In the tasks based on manual transcripts (T1 and T3) we submitted one run using the system described in Section 2.1. We refer to this system as QA$_m$. In the tasks based on automatic transcripts (T2 and T4) we submitted two runs: one using the system initially tailored for manual transcripts, QA$_m$, and another using the system tailored for automatic transcripts, where we used the PHAST keyword matching algorithm (see Section 4) and the NERC expanded with phonetic attributes (Section 3.2). We refer to the latter system as QA$_a$.

The corpora were pre-processed as follows. We deleted word fragment markers and onomatopoeias and discarded utterance information in manual transcripts (tasks T1 and T3). Speaker turns in the AMI corpus (tasks T3 and T4) were marked as sentence boundaries (this influences our answer ranking heuristics [9]) and the dialog was collapsed into a single document without speaker information. For the CHIL automatic transcripts (task T2) all non-word tokens were deleted (e.g., "{breath}") and utterance markers and fragment words were eliminated. Then the documents were pre-processed by a POS tagger, lemmatizer, and the NERC described in Section 3.

Table 1 summarizes our overall results. We report two types of scores: (a) TOP k, which scores a question as correct and assigns a score of 1 only if the system provided a correct answer in the top k returned (in this table we use $k = 1$, meaning that the correct answer must be returned on the first position to be considered); and (b) Mean Reciprocal Rank (MRR), which assigns to a question a score of $1/k$, where k is the position of the first correct answer, or 0 if no correct answer is returned. An answer is considered correct by the human evaluators if it is "exact", i.e., it contains the complete answer and nothing more, and it is supported by the corresponding document. If an answer was incomplete or it included more information than necessary the human assessors marked it as "non-exact". If an answer document did not provide the justification for the answer the answer was marked as "unsupported". In both these cases no credit was given to the QA system.

A first glimpse at the scores in Table 1 indicates that the results obtained are very encouraging: in five out of six of our submitted runs the TOP1 score was over the mean TOP1 score observed in TREC 2006 for factoid questions (0.18). In fact, for task T1 we obtain a score comparable with the top two best scores at TREC 2006 for factoid questions: 0.58 and 0.54. Arguably, the two evaluations are not directly comparable: both the question sets and the document collections are different. Nevertheless, the fact that our system obtains approximately the same performance on speech transcriptions as other, more complex systems on written text is proof that QA technology can be successfully used in speech-only scenarios.

Table 1 also shows that moving from manual transcripts to automatic transcripts (i.e., the difference between T1 and T2 scores, or T3 versus T4) yields a drop in TOP1 score of 0.15 in the CHIL collection and 0.04 in the AMI corpus. In relative terms, this is a drop of the TOP1

Table 2: Distribution of correct answers (TOP5) according to answer type. Org = organization, Per = person, Tim = time, Mea = measure, Met/Sys = method/system, Mat = material, Col = colour.

Task and System	Org	Per	Loc	Tim	Mea	Met/Sys	Lan	Sha	Mat	Col
T1, QA$_m$	10/20	8/9	4/9	7/10	12/28	10/18	3/4	-	-	-
T3, QA$_m$	2/13	0/15	1/14	2/14	4/12	-	1/2	5/9	4/6	8/11

score of 29% in CHIL and 16% in AMI. To our knowledge, this is the first time that such an analysis is performed for QA technology. Again, it is encouraging to see that, even when using the imperfect automatic transcripts, our scores are higher than the mean scores observed previously for written text. Somewhat surprisingly, the performance drop is smaller for the AMI corpus, even though these transcripts had a higher WER than the CHIL transcripts (38% versus 20%). The explanation is that, because the AMI tasks are harder due to the larger corpus and the more ambiguous question terms, we answer only the "easier" questions in the manual transcripts. Such questions tend to have a larger number of question keywords (i.e., a larger answer context) and answers that appear repeatedly in the collection, so the probability that the system encounter a valid answer even in automatic transcripts is large. In contrast, the CHIL corpus is very small, so one ASR mistake may be sufficient to lose the only existing correct answer for a given question. Based on these experiments, we can conclude that the QA performance drop more or less follows the WER in small corpora with little redundancy (e.g., CHIL) and is smaller than the WER in larger corpora where redundancy can be exploited (e.g., AMI).

One unexpected result in this evaluation was that the QA$_a$ system performed worse than the QA$_m$ system on automatic transcripts (tasks T3 and T4), even though the QA$_a$ system was designed to function with automatic transcripts. The explanation is two fold. First, with our current parameter setting, the PHAST algorithm triggered too many false keyword matches due to a relaxed approximated match. This yielded sets of candidate passages and answers with a lot of noise that was hard to filter out. Second, the NERC training data (i.e., the development corpus) was insufficient to learn correct phonetic generalizations, so many answer candidates were missed in automatic transcripts. In fact, in our experiments with the development corpus of automatic transcripts, the NERC with phonetic arguments performed the same as the one without phonetic information. Nevertheless, we believe that the architecture of the QA$_a$ system is a good long-term investment because it is the only one of the two systems developed that can address the phenomena specific to automatic transcripts.

Table 2 shows the distribution of correct answers according to the answer type for tasks T1 and T3. The table indicates that our system had a particularly hard time answering questions in task T3, where the answer type was a NE of type: `Per`, `Loc`, `Org`, or `Tim`. These entity types have a high variation in the AMI corpus and our NERC could not generalize well given the small amount of training data available (see also the error analysis below). This suggests that a better strategy for NERC would be to train an open-domain NERC, where large annotated corpora are available, and use domain transfer techniques to adapt the open-domain system to the AMI domain.

Finally, Table 3 summarizes the error analysis of the three system components: QP, PR, and AE. The "Questions" column lists the total number of questions in the corresponding task. The "QC Correct" column lists the number of questions with the answer type correctly detected by the question classifier (QC). The "PR Correct" column shows the number of questions where at least one passage with the correct answer was retrieved. The "QC & PR Correct" column lists the number of questions where the QC prediction is correct *and* PR retrieved a correct passage. Finally, the "TOP1" column shows the number of questions answered correctly with the exact answer on the first position. We can draw several important observations from this error analysis:

- The QC performs significantly worse for the CHIL question set (tasks T1 and T2) than the AMI questions. This suggests that one particularity of this evaluation was that the CHIL questions were more domain specific than the AMI questions.

Table 3: Error analysis of the QA system components.

Task and System	Questions	QC Correct	PR Correct	QC & PR Correct	TOP1
T1, QA$_m$	98	67	82	54	50
T2, QA$_a$	98	67	80	29	24
T2, QA$_m$	98	67	76	37	36
T3, QA$_m$	96	87	73	66	25
T4, QA$_a$	93	87	52	47	13
T4, QA$_m$	93	87	58	53	21

- PR performs similarly to the state of the art for written text for tasks T1, T2, and T3, but it suffers an important performance hit on task T4, where we processed automatic transcripts with the highest WER (38%). This proves that PR is indeed affected by a high WER.

- PR using PHAST performed better than the PR with exact keyword match for task T2 and worse for task T4. As previously mentioned, this worse-than-expected behavior of PHAST was due to the many false-positive keyword matches generated in our current setup. We leave the better tuning of PHAST for the various QA tasks as future work.

- For tasks T1 and T2, when the QA system reaches AE with the correct information (i.e., the "QC & PR Correct" column in the table), AE performed very well: we answered most of those questions correctly on the first position. This indicates that both the NERC and the answer ranking performed well. For tasks T3 and T4, the story is no longer the same: we suffer the biggest performance hit in AE. We inspected these errors post evaluation and the conclusion was that in most of the cases the fault can be assigned to the NERC, which failed to recognize the entity mentions that were correct answers in both manual and automatic transcripts. This problem was mitigated in tasks T1 and T2 with a combination of NERC models, which included a rule-based system that we developed previously for the CHIL domain [9].

6 Conclusions

This paper describes UPC's participation in the CLEF 2007 Question Answering on Speech Transcripts track. We were one of the few participants that submitted runs in all four sub-tasks and we obtained the highest overall score. Our best performing runs have TOP1 scores that range from 0.21 (on automatic transcripts with WER of 38%) to 0.51 (on manual transcripts). Both these scores are higher than the mean TOP1 score observed for factoid questions and written-text documents in TREC 2006 (0.18).

In this evaluation we analyzed the behavior of two systems. Both make minimal use of syntactic analysis (the document collection is only POS tagged) and both use a data-driven query relaxation algorithm to extract the best answer context from the input question. The difference between the two systems is that one is tailored for manual transcripts, i.e., it uses exact keyword matching in both PR and AE, while the other is tailored for automatic transcripts, i.e., is uses approximate keyword matching based on phonetic distances and deploys a NERC enhanced with phonetic features.

In all four sub-tasks we obtained the best performance with the system that was initially designed for manual transcripts. This system performed better than expected on automatic transcripts for two reasons: first, it only requires that the document collection be POS tagged, and POS tagging is a technology that is robust enough to function well on less-than-perfect automatic transcripts. Second, the query relaxation algorithm adapts well to automatic transcripts: question terms that are incorrectly transcribed are automatically discarded from the answer context. On the other hand, the system designed for automatic transcripts performed worse than expected because the approximated keyword match algorithm generated to many false-positive matches, which introduced to much noise in the candidate sets of passages and answers. Nevertheless, we believe that this approach is a good long-term research direction because it is the only one of the two systems developed that can truly address the phenomena specific to automatic transcripts.

Acknowledgements

This work has been partially funded by the European Commission (CHIL, IST-2004-506909) and the Spanish Ministry of Science and Technology (TEXTMESS project). Mihai Surdeanu is a research fellow within the Ramón y Cajal program of this ministry.

References

[1] S. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3, 1974.

[3] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, Massachusetts, 1979.

[4] B. Kessler. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103:243–260, 2005.

[5] G. Kondrak. *Algorithms for Language Reconstruction.* PhD thesis, University of Toronto, 2002.

[6] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Docklandy*, 10:707–710, 1966.

[7] X. Li and D. Roth. Learning question classifiers: The role of semantic information. *Journal of Natural Language Engineering*, 2005.

[8] M. Paşca. *High-performance, open-domain question answering from large text collections.* PhD thesis, Southern Methodist University, Dallas, TX, 2001.

[9] M. Surdeanu, D. Dominguez-Sal, and P. R. Comas. Design and performance analysis of a factoid question answering system for spontaneous speech transcriptions. *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2006)*, 2006.

[10] M. Surdeanu, J. Turmo, and E. Commelles. Named entity recognition from spontaneous open-domain speech. *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2005)*, 2005.