# Spoken Question Answering

Pere R. Comas i Umbert
pcomas@lsi.upc.edu

*Directors*
Jordi Turmo Borràs
Lluís Màrquez Villodre

Memòria del DEA i Projecte de Tesi
Programa de Doctorat en Intel·ligència Artificial
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Gener del 2008

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

The task of Question Answering (QA) consists in providing short, relevant answers to natural language questions. Most Question Answering research has focused on extracting information from text sources, providing the shortest relevant text in response to a question [45, 30]. For example, the correct answer to the question *"How many groups participate in the CHIL project?"* is *"16"*. Whereas the response to the question of *"Who are the partners in CHIL?"* is a list of the partners. This simple example illustrates the two main advantages QA has over current search engines: first, the input is a natural language question rather a keyword query, and second, the answer provides the desired information content and not a potentially large set of documents or URLs that the user must read through.

Current QA technology is focused mainly on the mining of written text sources for extracting the answer to questions both from open-domain and restricted-domain document collections. However, most human interaction occurs through spontaneous speech, e.g. meetings, seminars, lectures, telephone conversations, and are beyond the capacities of current QA systems. All these scenarios provide large amounts of information that could be mined by QA systems. As a consequence, the exploitation of spontaneous speech sources brings QA a step closer to many real world applications.

In addition, spontaneous speech transcriptions differ from classical written text in many aspects, and this makes QA for spontaneous speech transcriptions an interesting research area. The most common differences are:

- False starts and sentences interrupted in the discourse.
- The repetition of words. (e.g., *"I don't know where where the people will be"*)
- The use of onomatopoeias. (e.g., *"Rufford, um, Sanatorium, that's right."*)
- The lack of punctuation marks.
- The lack of capitalization.
- The presence of word errors due to the use of automatic speech recognizers (ASR). Typical errors are due to the lack of words in the language models (e.g., proper names in general), and the lack of representation in the acoustic model. In general, these errors are substitutions of word sequences for another ones (e.g., "feature" to "feather", "Barcelona" to "bars alone"), but never typo errors.

Consider the following example: one audio stream contains the information *"jacques chirac went to berlin"* and the user wants to know where the French president has been: *"Where did Jacques Chirac go?"*. If perfect transcriptions of the audio stream were available, this example would have an obvious solution and the whole problem would be no different than regular QA on written text (in this example there is no additional difference with well-written text). However, consider the case when the automatic transcription of the above stream contains two errors: *"went"* is transcribed as *"ate"* and *"berlin"* as *"barcelona"*. Hence the automatic transcription of the full stream is: *"jacques chirac ate to barcelona"*. In this case, the correct answer to be extracted is *"barcelona"*, because this is the text that points to the correct answer in the audio stream.

The grammatical structure of spoken language is different from that of written language, and some of the anchor points used in text processing such as punctuation must be inferred and are therefore error prone. There are disfluencies such as repetitions, restarts and corrections, and speech recognition errors. These differences explain why extracting answers from transcribed spontaneous speech requires more flexible QA architectures than those typically used for written text. Current techniques for text-based QA need substantial adaptation in order to access the information contained in audio data. This thesis project focuses on the proposal of new robust techniques for Question Answering on spoken documents.

## 1.2 Current and Future Work

During year 2007 we have involved in the organization of QAst, an international evaluation on QA over speech transcripts. It is the first event devoted to this topic to our knowledge.

The main part of this proposal covers the works undertaken to build a QA system for participate in QAst 2007 and the evaluation of the task. Our system is described in Chapter 4 and the QAst evaluation in Chapter 3.

The thesis proposal leans on our success in this evaluation, which points to it as a reliable research field.

In the immediate future we will organize and participate in QAst 2008 while exploring the research lines devised in Chapter 6.

## 1.3 Overview of this Document

This document is organized as follows. Chapter 2 reviews the state of the art in QA and general technologies for Information Retrieval applied to speech transcripts. Chapter 3 descrives the QAst evaluation framework. Chapter 4 explains our recent works on QA. It consist in building a robust QA engine to work with manual and automatic speech transcripts. Chapter 5 reports the organization of QAst, an international evaluation on QA over speech transcripts within the 2007 CLEF conference. It has been the main evaluation of our work. Chapter 6 states the thesis project and work plan for the future. It comprises the organization of QAst 2008 and 2009 and several new research lines. Finally, Chapter 7 presents the related publications.

# Chapter 2

# State of the Art

*"No existe el concepto del plagio: se ha establecido que todas las obras son obra de un solo autor, que es intemporal y es anónimo. La crítica suele inventar autores: elige dos obras disímiles —el Tao Te King y las 1001 Noches, digamos—, las atribuye a un mismo escritor y luego determina con probidad la psicología de ese interesante* homme de lettres*..."*

*Tlön, Uqbar, Orbis Tertius* - Jorge Luis Borges

Although there is a huge state-of-the-art in QA topic, there is no bibliography about spoken QA at all. The most similar scenario that can be found in literature is QA on spoken questions. But it is not an analogous task as we will show in Section 2.2.

Due to the novelty of this research field and this lack of specific literature, we have impulsed the creation of an international evaluation on QA over speech transcripts to help its development. The evaluation started in year 2007 within the Cross Language Evaluation Forum (CLEF[1]) and it is called QAst after Question Answering on Speech Transcripts. Section 2.3 reviews the approaches taken by the participants in QAst. This is an exhaustive review of all the literature on this topic.

Information Retrieval (IR) is a field strongly related to QA. In most of QA systems, IR is an intermediate step of the process. Section 2.4 reviews the literature on spoken document retrieval (SDR), which is much richer than spoken QA.

## 2.1 Classification of QA systems

Question answering systems can be classified from several perspectives. For example, the domain of its application (open domain or closed domain); the knowledge that it uses (e.g.

[1] `http://www.clef-campaign.org`

dictionaries, ontologies, very large knowledge bases); how the information is accessed (e.g. text documents, databases, Internet search engines).

International QA evaluations such as TREC, CLEF and NTCIR classify the results according to the types of questions that the systems must answer.

**Factoid questions:** Factoid questions can be answered with the name of an entity like persons, organizations, places, dates. Usually the answer is a noun of a noun phrase. These kind questions are also called *factual*.

**List questions:** These questions require a set of instances as the answer. Usually it is a closed list of entities such as from factoid questions. For example, the question *"Who are the partners in CHIL?"* is asking for is a list of organization names, this is a closed list.

**Definition questions:** Definitional questions are those looking for the definition of some entity or concept. Specially questions about the definition of some object or the biography of some person like in *"What is ELDA?"* or *"Who was Mervyn Peake?*

**Interactive questions:** In interactive questions, human and machine interact through a dialog system. The context of the dialog is taken into account, and the user can refine and clarify the questions and answers that is getting (also *follow-up questions*, FQ).

The scope of this thesis project is the factual question answering using spoken documents obtained by automatic transcription of spontaneous speech.

## 2.2 Question Answering

We won't attempt to draw an state-of-the-art of Question Answering for written text. This field is huge and a minimal part is addressed to spoken documents, which is an almost-new scenario. Detailed review of QA literature can be found, for example, in [7].

The only related works available are some attempts to build Voice-Activated Question Answering (VAQA) systems. VAQA focuses on answering spoken questions by integrating QA and automatic speech recognition in one pipeline. It is usually related to the interactive questions described in Section 2.1. Most of VAQA systems use an iterative refinement of voice and question processing taking advantage of the interactivity of the system [10, 39, 44]. Techniques specially designed for VAQA focus on checking the coherence of the ASR output bearing it must have the structure of a question. Possibly, the system may require the user to ask the question again or to clarificate some of its parts. Most of these systems focus on time performance since they are interactive. Due to the interactive nature and limited domain of VAQA, these techniques are not useful for spoken document QA since spoken documents can not be iteratively refined.

## 2.3 QAst

This section reviews the state of the art in spoken question answering. It focuses in the participants of 2007 QAst evaluation other than the UPC. The five groups participating in QAst are the following:

- CLT, Center for Language Technology, Australia.[2]
- DFKI, Deutsches Forschungszentrum für Künstliche Intelligenz, Germany.[3]
- LIMSI, Laboratoire d'Informatique et de Mécanique des Sciences de l'Ingénieur, France.[4]
- TOKYO, Tokyo Institute of Technology, Japan.[5]
- UPC, Universitat Politècnica de Catalunya.

All five systems implement a similar pipeline architecture: Initally, the collection of transcripts is pre-processed to fulfill the requirements of further steps such as segmentation, indexation and probably named entity recognition. The questions are processed in order to obtain the necessary information to retrieve passages of the documents that may contain the answer. Finally the answers are extracted from documents according differents methodologies.

The rest of Section 2.3 describes the systems of CLT, DFKI, LIMSI and TOKYO. The UPC system is reported in Chapter 4. Table 2.1 summarizes the main characteristics of each system given they share a similar architecture.

**Center for Language Technology, CLT** [22]

CLT participated using AnswerFinder, a modular question answering system developed focusing on shallow semantic representation of questions and text, specialy adapted for QAst. Due to the disfluences of spontaneous speech (Section 1.1), AnswerFinder does not use syntactic and graph-semantic information, its question answering strategy is entirely based on finding an selecting the right named entities.

First, the question is analyzed using hand-constructed patterns to get the type of the expected answer. Next, a subset of sentences from the documents is selected by the use of a relevance metric based on word ovelap with the question. Finally, the answers are extracted from the remaining sentences using the confidence of the NERC when tagging the entities.

CLT focuses in building a NERC module specific for automatic speech transcripts called AFNER. AFNER is based on machine learning (maximum entropy models) and combines three kinds of information:

1. Hand-crafted regular expressions for characteristic patterns of dates, monetary expressions, percentages, etc.
2. Gazetteers of locations, person names and organisations with 55,000 entries.
3. Contextual features extracted from the words and its context, such as capitalisation, presence of digits, prepositions, etc.

AFNER can assign multiple tags to one word, for the sake of recall, and can tag nested entities. The system was trained using a mix of the BBN corpus[6] and the AMI corpus (see Section 3.2 for its description).

[2] `http://www.clt.mq.edu.au`

[3] `http://www.dfki.de`

[4] `http://www.limsi.fr/tlp/`

[5] `http://www.furui.cs.titech.ac.jp`

[6] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T33`

CLT submited two runs for evaluation, clt1 and clt2. The sole difference is that clt2 makes no use of machine learning in the NERC.

**Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI** [26]

DFKI participated in QAst with an adaptation of their existing QA system for written text.

First the document collection is preprocessed. Documents are split in sentences using a splitter tool based on a maximun entropy. It uses a language model for written text. Then the sentences are annotated by a NERC. Finally the sentences are indexed using Lucene IR engine.[7]

The question answering process is as follows. An analyzer processes the question and outputs a query for the IR engine and the expected answer type (a named entity type). Then the system retrieves a set of documents (sentences) containing the query words and at least one named entity of the expected answer type. All entities of the correct type are considered answer candidates ant its relative frequency is computed as selection criterion.

NERC used by DFKI is combination of three different NERC components. On one hand, two free available NERC tools basen on statistical supervised learning, and on the other hand, another statistical NERC developed by DFKI that can recognize all the entity types used in QAst (see Section 3.2.2). All this NERC tools were trained on written text.

**Laboratoire d'Informatique et Mécanique des Sciences de l'Ingénieur, LIMSI** [33]

As a previous step, documents are analyzed. Case information and punctuation is reconstructed using an statistical language model estimated from the European Parliament Proceedings. Then the words are labeled with a mix of POS tags, chunk tags and name entity tags. Named entites are recognized using rules based on hand-made regular expressions of words. The analysis is multi-pass and enables the use of wold lists, sub-expressions, rule priorizing and other features. The rules where specialy crafted for speech.

The question answering process is as follows. An analyzer processes the question and ouputs an expectes answer type and a set of document queries obtained according the relative importance ordering of the entities in the question (e.g. named entity, noun, adjective...). The queries are submited to the indexation server until some passages are returned. LIMSI use two different strategies for query generation. In limsi1, the queries are generated by a large set of hand-made rules. In limsi2, the search is done by simmilarity between documents and a question descriptor (e.g. aranking of words).

There are also two strategies for answer extraction. In limsi1, the most frequent entity of the expected type is selected as the answer. In limsi2, an empirical formula uses keyword distance, frequency, passage retrieval confidence and question descriptor to rank the candidates and select the best one.

[7] http://jakarta.apache.org/lucene

**Tokyo Institute of Technology, TOKYO** [49]

Tokyo Institute of Technology uses an antirely data-driven, non-linguistic QA framework based on a noisy-channel model of the problem. It relies on data redundancy to help identify correct answers and no special adaptation was performed for QAst.

This system is composed of just two modules, one for information retrieval (IR) and one for answer extraction (AE). Both are statistical and its parameters must be specially tuned for every corpus. It makes no use of named entities neither any linguistic information.

IR employs a language modeling approach. The documents are ranked according to the conditional probability of generating the query $Q$ given the document $D$, $P(Q|D)$. Due to the scarcity of training data for QAst, a unigram model is applied. As a previous step, documents are segmented in sentences using an in-house segmenter.

The AE module models the probability of an answer $A$ given a question $Q$ as:

$$P(A|Q) = P(A|W, X),$$

where $W$ are the features describing the quiestion-type (e.g. *what*, *when*, *who*), while $X$ is a set of features describing what the question refers to.

TOKIO participated with two systems, tokyo1 and tokyo2, each one using different smoothing formulas in the IR module.

QAst results are reported in the overview document [43] and will be discussed in detail in Chapter 5. Roughly, LIMSI outperformed CLT, DFKI and TOKYO in all four traks of QAst. We think it is due to this is the only system in which all parts have been designed for robustness on speech. CLT and DFKI participated with adaptations of systems designed for written text, they had to neglect some of its modules and its performance was sensible reduced. The lack of adequated training data for NERCs based on machine-learning was another drawback. The pure statistical approach of TOKYO was also degraded by the scarcity of development data. It achieved worse results than in 2006 TREC factoid QA evaluation [48].

## 2.4 Spoken IR

Since affordable technology allows the storage of large masses of audio media, more and more spoken document sources become available to public access. This great body of spoken audio recordings is mainly unaccessible without accurate techniques of retrieval. Spoken document retrieval (SDR) is the task of retrieving passages from collections of spoken documents according to a user's request or query.

Classically, the approach to SDR problem is the integration of an automatic speech recognizer (ASR) with information retrieval (IR) technologies. The ASR produces a transcript of the spoken documents and these new text documents are processed with standard IR algorithms adapted to this task.

There is a vast literature on SDR for non spontaneous speech. For example, TREC conference had a spoken document retrieval task using a corpus composed of 550 hours of Broadcast News. TREC 2000 edition concluded that spoken news retrieval systems achieved almost the same performance as traditional IR systems [8]. Spontaneous speech contains disfluencies that can barely be found in broadcast news, such as repetition of words, the use of onomatopoeias, mumbling, long hesitations and simultaneous speaking. Little research has been done for spontaneous speech audio, like telephone conversations, lectures and meetings.

The following subsections give background information about ASR technology and IR methods used in SDR.

### 2.4.1 Automatic Speech Recognition

Given an acoustic signal the ASR searches for the most likely word sequence that could produce the signal when uttered. The ASR uses two statistical models for this task: an *acoustic model*, which relates signal and phones, and a *language model*, which estimates the probability of a certain sequence of words.

Given the input signal $A$, a word sequence $\widetilde{W}$ is generated according to the following rule:

$$\widetilde{W} = \arg\max_i P(A|W_i)P(W_i)$$

where $\widetilde{W}$ is the most likely word sequence, having the maximum *a posteriori* probability in the model. $P(A|W_i)$ is the probability of sequence $W_i$ sounds like $A$ (acoustic model) and $P(W_i)$ is the general probability of the sequence $W_i$ (language model). The most likely word sequence $\widetilde{W}$ is also called the *one-best* output. In fact, $A$ is a vector of acoustic features taken from the input signal. The dominant paradigm in speech recognition is the hidden Markov model (HMM). An HMM is a stochastic model, in which the generation of phoneme strings and words are represented probabilistically as Markov processes. HMM is a natural model since captures the temporal sequence of speech and its realization. More information on ASR and HMM can be found elsewhere [31].

Any ASR is limited in the size of the vocabulary it can recognize. This vocabulary depends on the amount of audio data used in its training. Therefore the ASR can not recognize all possible words, these are the so called Out of Vocabulary (OOV) words. The ASR transcribes the audio corresponding to these words as other sequence words in its closed vocabulary. Words such as proper names tend to be OOV and this is an additional difficulty for SDR since this words typically occur in user queries. OOV words may be very important for some applications as we will show lately.

### 2.4.2 Spoken Document Retrieval

Traditional text retrieval techniques assume the correctness of the words in the documents and the existence of structural information such as punctuation marks, paragraph boundaries, and capitalization. However, the speech recognition stage introduces errors that challenge traditional IR algorithms.

Nevertheless, results show that a reasonable approach to SDR consists in taking the one-best output of ASR (i.e., the most probable sequence of words that generates the input audio) and performing IR on this transcript. It works reasonably well when recognition is mostly correct and documents are long enough to contain correctly recognized query terms. This is the case of TREC 2000 evaluation on Broadcast News [8].

The Spoken Document Retrieval track in CLEF evaluation campaign uses a corpus of spontaneous speech for cross-lingual speech retrieval (CL-SR) [47, 27, 29]. CL-SR corpus is the Malach corpus, which is composed of nearly 600 hours of spontaneous speech from interviews with Holocaust survivors. This is a more general scenario than former TREC tracks.

During the CL-SR overview discussion at CLEF 2007 workshop was stated that one of the main reasons to end the track was that SDR is almost useless for a real-world application. There is a significant difference between time and the effort necessary to read through a list of text snippets and listening to a list of audio excerpts. It is critical for the human user to get an output so short and precise that its beyond actual possibilities. From this point of view, QA is much more useful than SDR since it provides exact answers and it is human affordable. Note that a human user do need to listen to the original audio recording since the transcript may contain errors and lacks of punctuation.

Approaches to SDR can be classified in two categories according to their use of ASR–specific data. Some methods only use the one–best output as is, therefore it is independent of the specific ASR characteristics. Other methods take advantage of additional information supplied by the ASR. Some ASRs may output additional information (it depends on its implementation) such as confidence scores, $n$–best output, full lattices. The use of this information or other ASR–error models makes dependant of a concrete ASR.

### ASR Independent Retrieval

Most of participants in TREC and CL-SR evaluations use ASR independent methods since no additional ASR information is available.

Recent results show that a reasonable approach to SDR consists in taking the one-best output of ASR (i.e., the most probable sequence of words that generates the input audio) and performing IR on this transcript. It works reasonably well when recognition is mostly correct and documents are long enough to contain correctly recognized query terms.

Top ranked participants in CL-SR, see [2, 15, 12, 46], used a wide range of traditional text based IR techniques. Good results were achieved with term-based ranking schemes such Okapi BM25 [32], Divergence From Randomness [3] and Vector Space Models [34]. Most of the work done by the participants was focused on investigating the effects of meta-data, hand-assigned topics, query expansion, thesauri, side collections and translation issues. Some participants used $n$-gram based search instead of term search. For $n$-gram search, text collection and topics are transformed into a phonetic transcription, then consecutive phones are grouped into overlapping $n$-gram sequences, and finally they are indexed. The search consists in finding $n$-grams of query terms in the collection. Some experiments show how phonetic forms helps to overcome recognition errors. Some results using phonetic $n$-grams are reported in [13] showing only slightly improvements.

### ASR Dependant Retrieval

Experimental results show that the traditional approach consisting of ASR and IR is not useful if the task requires the retrieval of short speech segments in a domain with higher word error rate. In this cases, other approaches to SDR have been proposed. Most try to improve retrieval performance using additional information specific to the ASR. For example, Srinivasan and Petkovic [38] use an explicit model of the ASR error typology to address the OOV problem. First, they use two ASRs to generate a word transcript and a phonetic transcript of the input audio. Then they build a phone confusion matrix that models the probability of ASR mistaking any phone for a different one. Finally, the retrieval step uses a Bayesian model to estimate the probability that the phonetic transcript of a speech segment is relevant to the query term.

Another common approach is the use of ASR lattices to make the system more robust to recognition errors. The lattice is an internal ASR data structure which contains all possible outputs given the audio input. For example, experiments in [36] report an improvement of 3.4% in $F_1$ measure in Switchboard corpus using a combination of word-lattices and phone-lattices as search space. The use of word-lattices alone cannot overcome the problem of OOV words.

| System | Enrichment | Question classification | Doc/Pass Retrieval | Answer Extraction | NERC |
|---|---|---|---|---|---|
| clt1 | words and NEs | hand-crafted patterns | passage ranking based on word similarities between passage and query | candidate ranking based on frequency and the NER confidence | hand-crafted patterns, gazetteers and ME models |
| clt2 | | | | | no ME models |
| dfki1 | words and NEs | hand-crafted<br>sintactic-sematic rules | Lucene IR | candidate ranking based on frequency | gazetteers and not tuned statistical models |
| limsi1 | words and NEs | hand-crafted patterns | pass. ranking based on hand-crafter back-off queries | candidate ranking based on frequency, keyword distance and retrieval confidence | hand-crafted patterns |
| limsi2 | | | cascaded doc/pass ranking based on search descriptors | | |
| tokyo1 | words | non-linguistic statistical multi-word model | pass. retrieval with interpolated doc/pass statistical models | candidate ranking based on statistical multi-word model | no |
| tokyo2 | | | addition of word classes to the statistical models | | |
| upc1 | words, NEs lemmas and POS | perceptrons | pass. ranking based on iterative query relaxation | candidate ranking based on keyword distance and density | hand-crafted patterns, gazetteers and perceptrons |
| upc2 | also phonetics | | addition of approximated phonetic matching | | |

Table 2.1: Summary of the characteristics of systems participating in QAst

# Chapter 3

# Evaluation Framework QAst

*"No hables por los codos. Aumentas el caos."*

*La mirada de la condesa Hahn-Hahn* - Péter Esterházy

During year 2007 we have been involved as coordinators in the organization of an international evaluation on QA over speech transcripts within the Cross Language Evaluation Forum (CLEF).[8] To our knowledge, this is the first event devoted to this topic CLEF fosters research and evaluation of cross-language information retrieval focusing on European languages.

We have organized a new evaluation track called QAst (Question Answering over Speech Transcripts) which provides a testbed for our research and to other people interested in speech transcripts. The 2007 edition is a pilot track and it will last at least until CLEF 2009. In these three editions we hope to explore new QA issues regarding language, word error rate, and speaker context. Information about future plans for QAst 2008 is given in 6.1 or in the official web.[9] QAst has been organized together with LIMSI[10] and ELDA.[11]

The rest of this chapter summarizes the QAst 2007 evaluation framework.

## 3.1 The QAst task

Existing evaluation frameworks do not evaluate factual QA systems for oral transcripts. This pilot track aims at providing a framework in which QA systems can be evaluated when the answers have to be found in spontaneous speech transcripts (manual and automatic transcripts). There are three main objectives to this evaluation:

- Comparing the performances of the systems dealing with both types of transcripts.

[8] http://www.clef-campaing.org

[9] http://www.lsi.upc.edu/~qast

[10] http://www.limsi.fr/tlp/

[11] http://www.elda.org

- Measuring the loss of each system due to the inaccuracies in state of the art ASR technology.
- Motivating and driving the design of novel and robust factual QA architectures for automatic speech transcripts.

In this evaluation, the QA systems have to return answers found in the audio transcripts.

**Answer:** : it is the minimal sequence of words that includes the correct exact answer to the question in the audio stream.

For the purposes of this evaluation, instead of pointers in the audio signal, the recognized words covering the location of the exact answer have to be returned. For example, consider the question *"Which organization has worked with the University of Karlsruhe on the meeting transcription system?"*, and the following extract of an automatically recognized document:

```
{breath} {fw} and this is joint work between University of Karlsruhe and
coming around so {fw} all sessions once you find {fw} like only stringent
custom film canals communicates on on {fw} tongue initials
```

corresponding to the following exact manual transcript:

```
uhm this is joint work between the University of Karlsruhe and Carnegie
Mellon, so also here in these files you find uh my colleagues and uh Tanja
Schultz.
```

The answer found in the manual transcript is *Carnegie Mellon* whereas in the automatic transcript it is *coming around*. This example illustrates the two principles that guided this track:

- The questions are generated considering the exact information in the audio stream regardless of how this information is transcribed, because the transcription process is transparent to the user.
- The answer to be extracted is the minimal sequence of words that includes the correct exact answer in the audio stream (i.e., in the manual transcripts). In the above example, the answer to be extracted from the automatic transcript is *coming around*, because this text gives the start/end pointers to the correct answer in the audio stream.

## 3.2 QAst Setting

### 3.2.1 Data collections

The data for the QAst pilot track consists of two different resources, one for dealing with the lecture scenario and the other for dealing with the meeting scenario:

- The CHIL corpus:[12] consists of around 25 hours (around 1 hour per lecture) both manually and automatically transcribed (LIMSI[13] produced the ASR transcriptions with

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

[13] http://www.limsi.fr/tlp/

| Type | Examples |
|---|---|
| person | Stefan Kantak, Maria Danninger, Phil |
| organization | AT&T, Carnegie Mellon |
| location | Dallar, Texas, USA, Geneva |
| time | two thousand and two, january, thursday twenty-four |
| measure | twnety-five point zero percent, ten grams |
| method | flexible three clustering, |
| system | Emacs, Java |
| language | english, french |
| color | red, green, blue |
| shape | triangle, banana |
| material | silver, plastic, fiberglass |

Table 3.1: Examples of the NEs

around 20% of word error rate [19], while the manual ones were done by ELDA[14]). In addition, the set of lattices and confidences for each lecture has been provided. The domain of the lectures is *speech and language processing*. The language is European English (mostly spoken by non native speakers). Seminars are formatted as plain text files [23].

- The AMI corpus:[15] consists of around 100 hours (168 meetings) both manually and automatically transcribed (the University of Edinburgh produced the ASR transcripts with around 38% of WER [9]). The domain of this meetings is *design of television remote control*. The language is European English. Meetings are formatted as plain text files.

Four tasks have been defined for QAst:

- 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.

### 3.2.2 Questions and answer types

All the questions in the QAst task are factual questions, whose expected answer is a Named Entity (NE). The entities are: person, location, organization, language, system, method, measure, time, color, shape and material. Some of this entity types, such as color, shape ans system, does not belong to the classical types of TREC and CLEF and are new for this evaluation. No definition questions have been proposed. Table 3.1 summarizes the different NEs.

For each one of the scenarios, two sets of questions were provided to the participants:

[14] http://www.elda.org

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

- Development set:
  - T1/T2: 10 seminars and 50 questions.
  - T3/T4: 50 meetings and 50 questions.
- Evaluation set:
  - T1/T2: 15 seminars and 100 questions.
  - T3/T4: 118 meetings and 100 questions.

Both data collections were first tagged with Named Entities. Then, an English native speaker created questions for each NE tagged session. So each answer is a tagged Named Entity. An small percentage of the questions doesn't contain an answer in the documents.

An answer is basically structured as an [answer-string, document-id] pair, where the answer-string contains nothing more than a complete and exact answer (a Named Entity) and the document-id is the unique identifier of a document that supports the answer. There were no particular restrictions on the length of an answer-string (which is usually very short), but unnecessary pieces of information were penalized, since the answer could be marked as non-exact. Assessors focused mainly on the responsiveness and usefulness of the answers.

### 3.2.3 Human judgement

The files submitted by participants were manually judged by native speaking assessors. Assessors considered correctness and exactness of the returned answers. They have also checked that the document labelled with the returned document-id supports the given answer. One assessor evaluated the results. Then, another assessor manually checked each judgement evaluated by the first one. Any doubts about an answer were solved through various discussions.

For T2 and T4 (QA on automatic transcripts) the manual transcriptions were aligned to the automatic ASR outputs to find the answer in the automatic transcripts. The alignments between the automatic and the manual transcription were done using time information for most of the seminars and meetings. Unfortunately time information was not available for some AMI meetings and only word alignments were used.

The four possible judgements (also used at TREC [45]) correspond to a number ranging between 0 and 3:

- 0 correct: the answer-string consists of the relevant information (exact answer), and the answer is supported by the returned document.
- 1 incorrect: the answer-string does not contain a correct answer or the answer is not responsive.
- 2 non-exact: the answer-string contains a correct answer and the document-id supports it, but the string has bits of the answer missing or is longer than the required length of the answer.
- 3 unsupported: the answer-string contains a correct answer but the document-id does not support it.

### 3.2.4 Measures

The participants could give a ranked list of 5 possible answers for each question. The two following metrics used in CLEF have been used in the QAst evaluation:

1. Mean Reciprocal Rank (MRR) measures how well ranked is the right answer. MRR is the averaged multiplicative inverse of the rank of the correct answer in the answers list. Each question scores $1/k$, where $k$ is the position of the first correct answer in the list, or 0 if no correct answer is returned.
2. Accuracy: The fraction of correct answers ranked in the first position in the list of 5 possible answers.

A question is correct if it is "exact", i.e., it contains the complete answer and nothing more, and it is supported by the corresponding document.

# Chapter 4

# Architecture

*"Would James Joyce and De Selby combine their staggeringly complicated and diverse minds to produce a monstrous earthquake. . . ?"*

*Dalkey Archive* - Flann O'Brien

In this chapter we describe our proposal of architecture for QA on speech transcripts. Section 4.1 summarizes the objectives that guide our proposal, Section 4.2 is an overview of the architecture, and Sections 4.3 and 4.4 focuses on the NERC (Named Entity Recognition and Classification) and Passage Retrieval tasks.

Our main contributions are:

- A robust Answer Extraction (AE) method. (Section 4.2.2)
- The development of a NERC (Named Entity Recognition and Classification) specific for speech data. (Section 4.3.2)
- A new Passage Retrieval (PR) algorithm designed for speech transcripts. (Section 4.4)

Chapter 5 reports the results of this proposal obtained in the QAst evaluation along with QAst general results.

## 4.1 Task Description

Most of our architecture is designed specifically for the QAst evaluation (see 3.1 for further details). QAst consists of the following four tasks:

**T1** : QA using as underlying document collection the manual transcripts of the lectures.

**T2** : QA using the automatic transcripts of T1 lectures.

**T3** : QA in the manual transcriptions of the meetings

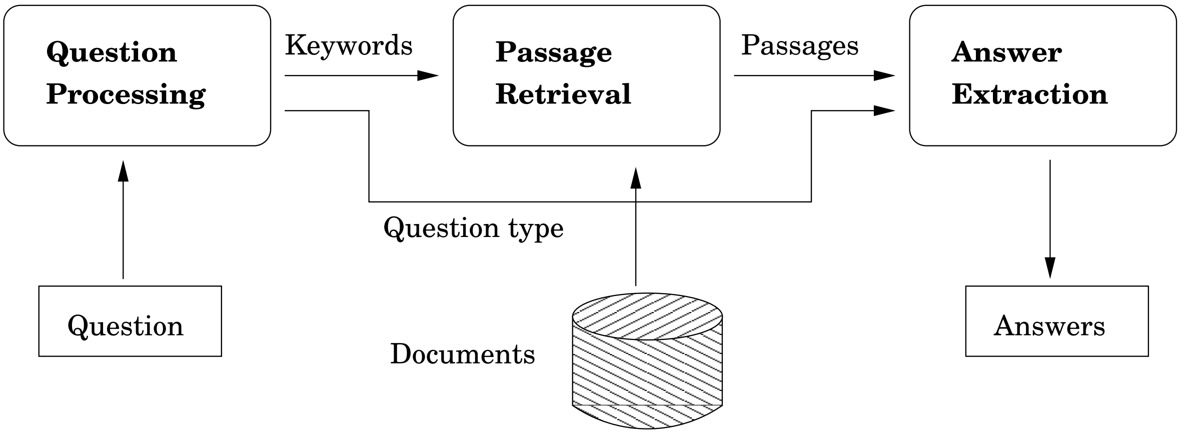

Figure 4.1: High-level architecture.

**T4** : QA in the automatic transcripts of the T3 meetings.

For tasks T1 and T3 we have adapted a QA system and NERC that we previously developed for the processing of manual speech transcripts [40, 42]. 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 handling 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 phonetic 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.

## 4.2 General 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, as is shown in figure 4.1. The next sub-sections 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.

### 4.2.1 QA System for Manual Transcripts

For the processing of manual transcripts we used an improved version of the system introduced in [40]. We describe it briefly below.

**Question Processing (QP)** 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 [21][16] and 3 additional types that are specific to

[16] This is a frequently used resource.
Data collection is donwloadable from `http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/`

the corpora used in this evaluation (i.e., `system/method`, `shape`, and `material`). The answer types are predicted using a multi-class Perceptron classifier and a rich set of lexical, syntactic (part-of-speech tags and syntactic chunks) and semantic (i.e., distributional similarity) features . This classifier obtains an accuracy of 88.5% on the corpus of [21]. 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 (PR)** The goal of this component is to retrieve a set of relevant passages from the document collection, given the previously ranked question keywords. The PR algorithm uses a dynamic query relaxation (DQ) procedure that iteratively adjusts the number of keywords used for retrieval and their proximity until the quality of the recovered information is satisfactory ([40, 28]). In each of this iterations, a Document Retrieval application[17] 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.

DQ algorithm is shown in Algorithm 1. Where the set of keywords $K$ is initialized with all keywords with rank above the rank of verbs, and the current proximity is initialized with some default value (20 words in our experiments). The algorithm is configured with four parameters: $MinPass$ and $MaxPass$ – lower and upper bounds for the acceptable number of passages (currently 1 and 50), $MinProx$ and $MaxProx$ – lower and upper bounds for keyword proximity (currently 20 and 60 words). This algorithm uses limited (only POS tags) which makes it very robust for speech transcripts.

Figure 4.2 shows an example of passage construction for a simple query and one sample sentence.

**Answer Extraction (AE)** 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 seven heuristics that measure properties of the keywords and their context.

**H1** *Same word sequence* - computes the number of words that are recognized in the same order in the answer context.

**H2** *Punctuation flag* - 1 when the candidate answer is followed by a punctuation sign, 0 otherwise.

**H3** *Comma words* - computes the number of question keywords that follow the candidate answer, when the later is succeeded by comma. A span of 3 words is inspected. The last two heuristics are a basic detection mechanism for appositive constructs, a common form to answer a question.

**H4** *Same sentence* - the number of question words in the same sentence as the candidate answer.

**H5** *Matched keywords* - the number of question words found in the answer context.

[17] We have used Lucene IR engine: `http://jakarta.apache.org/lucene`

**Algorithm 1**

**DQ algorithm**

**Parameter:** $K$, keyword set

**Parameter:** $p$, proximity

```
1:  Retrieve passages using current $K$ and $p$
2:  if number of passages $< MinPass$ then
3:      if $p < MaxProx$ then
4:          increment $p$; goto 1
5:      else
6:          reset $p$
7:          drop the least-significant keyword from $K$; goto 1
8:      end if
9:  else if number of passages $> MaxPass$ then
10:     if $p > MinProx$ then
11:         decrement $p$; goto 1
12:     else
13:         reset $p$
14:         add the next available keyword to $K$; goto 1
15:     end if
16: end if
17: Return the current set of passages
```

**H6** *Answer span* - the largest distance (in words) between two question keywords in the given context. The last three heuristics quantify the proximity and density of the question words in the answer context, which are two intuitive measures of answer quality.

**H7** *Distance from QFW* - measures the distance between the candidate answer and the QFW (question focus word). The QFW, which indicates the question emphasis, is usually the head of the first noun or verb in the question, skipping stop words, auxiliary and copulative verbs. For example, in the question *"What is the Translanguage English Database also called?"*, the QFW is "database". This heuristic is enabled only for questions of numeric type, where typically the QFW appears very close to the answer. For example, the answer to the question: "How many stories does the tower of Pisa have?" is "8 stories".

All these heuristics can be implemented without the need for any NLP resources outside of a basic tokenizer. For each candidate answer, these seven values are then converted into an overall answer score using the formula below:

$$\textit{score} = \mathbf{H1} + \mathbf{H2} + 2\mathbf{H3} + \mathbf{H4} + \mathbf{H5} - \frac{1}{4}\sqrt{\mathbf{H6}} - \mathbf{H7}.$$

where the heuristic weights were previously optimized for a set of 200 questions. See [28] and our previous work in [40]. The above score drives the answer ranking that is finally reported to the user.

**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 4.2: 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 4.3: Examples of manual (M) and automatic (A) transcripts.

### 4.2.2 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 4.3 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 words). This increases the probability that the ASR incorrectly recognizes the named entities relevant for the AE component.

In order to address these specific issues 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 4.4.

## 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.

Our NERC implements the recognition and classification of named entities in a single sequential-tagging process, using the Begin-Inside-Outside (BIO) representation. Each token is classified into a BIO class using the averaged multiclass Perceptron of Crammer and Singer [6]. Features include lexical, syntactic (POS labels), and gazetteer information extracted from the current token and a context of two tokens to the left and to the right. Dynamic information ( i.e., the labels assigned to the previous token) is also extracted from the left context. A globally-consistent solution (e.g., no entity should start with an Inside label) is obtained using local greedy inference based only on the previously-labeled token.

### 4.3.1 NERC for Manual Transcripts

Our initial idea for the identification of NEs in manual transcripts was to use the NERC module we developed previously for the processing of manual speech transcripts [42]. One change from the previous system is that, to speed up training, we replaced the existing SVM classifiers with a multi-class Perceptron as stated previously.[18]

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 [40], (b) the above NERC trained on the existing CHIL development data, and (c) the above NERC trained on the CoNLL English corpus[19]. 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.

### 4.3.2 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 large maintained in the transcript.

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

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

For example, in Figure 4.3 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 [4]. The cluster identifier of each token is then added as a feature in the NERC model. For example, *"Sun"* and *"some"* share the same cluster identifier, 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.

## 4.4 The Phonetic Sequence Alignment Algorithm

This section shows how an approximated search based on phonetic similarity can be efficiently implemented using a sequence alignment algorithm drawn from the field of bioinformatics.

### 4.4.1 PHAST Algorithm

The recent sequencing of a large number of genomes has greatly stimulated the development of computational methods for the identification of patterns in biological sequences. Hence a family of pattern-matching algorithms for sequence comparison (sequence homology of proteins) in large databases has recently been developed. The most succeeding algorithm in the field is called BLAST (Basic Local Alignment Search Tool) [1]. BLAST employs a measure based on well-defined mutation scores to find regions of local similarity in protein sequences. BLAST is a simple and robust algorithm that can be applied in a variety of contexts.[20]

Following the approach of BLAST, we have implemented PHAST, an IR-engine over large phonetic sequences based on the same principles. Our hypothesis is that it is possible to find the best matchings of the keywords by searching for small contiguous substrings of phones (hooks) in the transcript, extending them and computing its relevancy.

PHAST algorithm has some advantageous properties for dealing with SDR: finds approximated matchings with independence of subword length, it can easily split/merge sequences, and no training data is required. In PHAST, this process is language independent given an appropriated set of phones.

Algorithm 2 shows a general view of PHAST scheme. It is a two-step process: First, term frequency is calculated using phonetic similarity, and second, a standard document ranking process takes place.

The input data is a collection of documents transcribed into phonetic sequences $\mathcal{D}$, and a set of keywords phonetically transcribed $\mathcal{KW}$. There are three important functions to describe in this algorithm.

- $detection_{\phi}(w, d)$ is a function that detects hooks within document $d$ considering keyword $w$. Different functions $\phi$ can be used to detect hooks as will be discussed in Section 4.4.2.
- $extension_{\varphi}(w, d, h)$ is a function that extends hook $h$ and computes an accurate similarity score $s$ between keyword $w$ and document $d$ around $h$. Different functions $\varphi$ can be used as will be discussed in Section 4.4.2.

[20] `http://www.ncbi.nlm.nih.gov/BLAST/`

**Algorithm 2**

**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 $h = detection_\phi(w, d)$ do
      $s = extension_\varphi(w, h)$
      if $relevant(s, h)$ then
         update $tf(w, d)$
      end if
   end while
end for
Rank collection $\mathcal{D}$
```

Reference transcript 3M: *"The host system it is a UNIX Sun workstation"*
Automatic transcript 3A: *"that of system it is a unique set some workstation"*

| | jun | ← $detection_\phi$ | |
|---|---|---|---|
| . . . ðæt ʌβ sɪstəm ɪt ɪz ə | jun | ik sɛt sʌm | wəʊrksteɪʃən. . . |
| | junik s sʌn | | ← $extension_\varphi$ |

Figure 4.4: Search of term "UNIX-Sun"

- $relevant(s, h)$ judges how this occurrence of $w$ at $h$ with score $s$ is relevant enough for term frequency. It triggers an update procedure relative to the document ranking process to be used. This will be discussed in Section 4.4.2.

Finally documents in $\mathcal{D}$ are ordered according to a ranking measure.

Figure 4.4 shows an example of how functions $detection_\phi$ and $extension_\varphi$ are used. Document $d$ is the sentence 3A from Figure 4.3, which has been transcribed to a sequence of phones. The query word $w$ is the term *"UNIX-Sun"*, which is transcribed as [juniks sʌn]. Term $w$ exists in the manual transcript 3M but not in the automatic transcript 3A. In the first step, $detection_\phi$ finds hook [junik] related to [juniks sʌn]. In the second step, $extension_\varphi$ extends the hook by matching the rest of [juniks sʌn] with the phones surrounding [junik] in the sentence.

### 4.4.2 Keyword search

This section describes the three functions $detection_\phi$, $extension_\varphi$ and *relevant*.

#### Detection of Hooks

In order to efficiently detect the occurrences of a phonetic sequence $a$ in a phonetic sequence $b$, $detection_\phi(a, b)$ is sequentially computed over the corpora. Function $\phi$ has been implemented following an approach similar to the one presented by Altschul [1]. Given a set of phonetically

| State | ə | a | ɪ | l | m | n | t | * |
|---|---|---|---|---|---|---|---|---|
| 0 | 15 | 6 | 10 | 4 | 3 | 1 | 0 | 0 |
| 1 | 15 | 6 | 10 | 4 | 2 | 1 | 0 | 0 |
| 2 | 15 | 6 | 13 | 4 | 3 | 1 | 0 | 0 |
| 3 | 15 | 6 | 19 | 4 | 3 | 1 | 0 | 0 |
| 4 | 15 | 5 | 10 | 4 | 3 | 1 | 0 | 0 |
| 5 | 15 | 6 | 18 | 4 | 3 | 1 | 0 | 0 |
| 6 | 15 | 6 | 7 | 4 | 3 | 1 | 0 | 0 |
| 7 | 15 | 6 | 10 | 4 | 3 | 8 | 0 | 0 |
| +8 | 15 | 6 | 10 | 4 | 12 | 1 | 9 | 0 |
| +9 | 15 | 6 | 10 | 4 | 3 | 1 | 0 | 0 |
| 10 | 15 | 6 | 10 | 4 | 3 | 11 | 0 | 0 |
| 11 | 15 | 6 | 10 | 4 | 12 | 1 | 9 | 0 |
| +12 | 15 | 6 | 13 | 4 | 3 | 1 | 0 | 0 |
| +13 | 15 | 6 | 10 | 4 | 3 | 14 | 0 | 0 |
| +14 | 15 | 6 | 10 | 4 | 12 | 1 | 9 | 0 |
| 15 | 15 | 6 | 10 | 16 | 3 | 1 | 0 | 0 |
| 16 | 15 | 17 | 10 | 4 | 3 | 1 | 0 | 0 |
| +17 | 15 | 6 | 18 | 4 | 3 | 1 | 0 | 0 |
| +18 | 15 | 6 | 10 | 4 | 3 | 8 | 0 | 0 |
| 19 | 15 | 6 | 10 | 4 | 3 | 14 | 0 | 0 |

Table 4.1: Transition table of a DFA recognizing all the 3-grams in [əlaɪnmɪnt]. Final states are marked with '+'. Initial state is 0

transcribed keywords, a deterministic finite automaton [11] DFA$_k$ is automatically built for each keyword $k$ in order to recognize all its possible substrings of $n$ phones.

For instance, given $n = 3$ and the keyword "alignment", which is phonetically transcribed as [əlaɪnmɪnt], there are seven phonetic substrings of length three (3-grams): əla, laɪ, aɪn, ɪnm, nmɪ, mɪn and ɪnt. One DFA is automatically built to recognize all seven 3-grams at once. Table 4.1 shows the transition table of this DFA.

Using these DFAs, the collection is scanned once to search for all the hooks. When a hook is found, a process for extending it is executed. This process is described in the following section.

**Extension of Hooks**

After a hook is found, PHAST uses $\varphi$ to extend each hook $h$ and to compute its score value $s$. Following the approach of Altschul we have taken the edit distance (Levenshtein distance [20]) as $\varphi$ function. Recent works have successfully used edit distance as a measure of phonetic similarity.

The process is as follows: when a hook $h$ is found, the edit distance is calculated between the query term $w$ and a subsequence of the document $d$ around the position where $h$ was found. This yields an score for the similarity of this occurrence of $w$ in $d$.

Given two sequences $a$ and $b$ of lengths $n$ and $m$, the edit distance algorithm can be computed by means of a dynamic programming algorithm as follows. In a first step, a distance matrix $D$ is computed. $D[i,j]$ holds the minimal distance between the prefixes of lengths $i$ and $j$ of sequences $a$ and $b$, respectively. Initially, the first row and first column of the matrix are filled with a multiple of the *indel* cost. Then each new element in $D$ is calculated recursively for longer substrings

$$\begin{aligned} D[i,j] = min( & D[i-1,j-1] + \delta(a_i,b_j), \\ & D[i-1,j] + \delta(a_i,-), \\ & D[i,j-1] + \delta(-,b_j)\ ) \end{aligned}$$

where $\delta(x,y)$ is the cost of substitute symbol $x$ for symbol $y$, $\delta(a_i,-)$ states the cost of deleting symbol $a_i$. The complexity of building this matrix is $\mathcal{O}(m \cdot n)$.

In a second step, the optimal alignment is retrieved from $D$. It starts at $D[n,m]$ and tracks back the high-scoring path until $D[0,0]$ is reached. The next entry of $D$ after the current one depends on the choice made in the first step. As a result, $D[n,m]$ is the total similarity score $\Delta(a,b)$. The complexity of the reconstruction step is $\mathcal{O}(m+n)$.

*The similarity function* $\Delta$

PHAST finds optimal alignment of sequences of phonemes using the edit distance with a phonetically motivated cost function.

Phonetic similarity functions have been used in other domains of research with success (e.g., identification of confusable drug names, dialectometry, spelling correction). A survey on phonetic similarity functions can be found in [16]. Kondrak [17, 18] proposes a flexible and mathematically sound approach. This approach uses the edit distance with two straightforward modifications of the original algorithm. The first one is the use of two new operations: compression and expansion. The second one is the use of a modified reconstruction step that allows global and semi-local alignment. These modifications are described below.

- Compression/expansion: Two contiguous symbols of one string may correspond to a single symbol of the other string. Compression and expansion are the same operation from a computational point of view. Compression allows better detection of phenomenons like consonant merging (e.g. [c] sounds like the pair [tʃ] rather than [t] or [ʃ] alone).
- Semi-local alignment: As described before, we have seen that the reconstruction step of the edit distance starts in $D[n,m]$ and builds the best *global* alignment of $a$ and $b$. If we try to align a short sequence (a keyword) with a longer sequence (a paragraph or sentence), we don't want to scatter the keyword across multiple words even if it involves no substitutions (we will have always a big amount of *indels* in this scenario) since it won't make sense from an empirical point of view. The effect of the alignment is depicted in Figure 4.5. The "semi-local" alignment has a better scoring around [gwɪst] than the traditional "global" alignment around [lɪŋgwɪst], but it is prefearable to keep the keyword phones together having the most of *indels* after or before the keyword. Although similarity between [f] and [l] is greater than [f] and [g], semi-local alignment gets a better score since it penalizes the four *indels* at [ɪŋgw] because they are within the match of [fɪst]. Therefore semi-local alignment is biased towards keeping together keyword phones.

Global alignment

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| f | - | - | - | - | ɪ | s | t | - | - | - |
| l | ɪ | ŋ | g | w | ɪ | s | t | ɪ | k | s |

Semi-local alignment

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| - | - | - | f | - | ɪ | s | t | - | - | - |
| l | ɪ | ŋ | g | w | ɪ | s | t | ɪ | k | s |

Figure 4.5: How global and semi-local affects the alignment of the phonetic transcription of words "fist" and "linguistics"

This behavior may be achieved modifying the initialization and reconstruction steps as described in [18].

Kondrak also proposes a metric for measure inter-phoneme similarity based on multi-valuated features with salience coefficients. Each phoneme is described from a physical point of view with multi-valuated features (e.g. articulatory point, roundness, etc.).

Tables 4.2 and 4.3 show the phones and features used by our system. The number enclosed in parenthesis is the numerical value of the feature. These features and values are based on those used in [18] and enhanced to deal with extra sounds from Spanish and Catalan languages. Table 4.4 shows the salience of the features used in our system, which are the same as in [18].

The $\delta(x, y)$ function used by Kondrak is the following:[21]

$$\begin{aligned}
\delta(a_i, -) &= k_1 \\
\delta(a_i, b_j) &= k_2 - \delta'(a_i, b_j) - V(a_i) - V(b_j) \\
\delta(a_i a_{i+i}, b_j) &= k_3 - \delta'(a_i, b_j) - \delta'(a_{i+1}, b_j) - \\
&\quad V(b_j) - max(V(a_i), V(a_{i+1}))
\end{aligned}$$

where $k_1$ is the cost of deleting a symbol, $k_2$ is the base score when matching two equal symbols, $k_3$ is the base score when compressing two symbols into one, and $k_4$ is a penalty for matching vowels with consonants, $\mathit{diff}(a, b, f)$ evaluates if feature $f$ is different in $a$ and $b$, and finally

$$\begin{aligned}
V(a_i) &= \begin{cases} 0 & \text{if } a_i \text{ is a consonant} \\ k_4 & \text{otherwise} \end{cases} \\
\delta'(a_i, b_j) &= \sum_{f \in features} \mathit{diff}(a_i, b_j, f) \cdot salience(f)
\end{aligned}$$

*The scoring function*

The edit distance described above provides a scoring procedure for extension of hooks. The greater the edit distance between a keyword and an extension of a hook, the lower the score value.

[21] $\delta(x, y)$ function is symmetric. For the sake of simplicity, just one direction is presented.

| Ph. | SRL | Place | High | Back |
|---|---|---|---|---|
| a | + - - | velar (0.6) | low (0.0) | front (1.0) |
| ɑ | + - + | velar (0.6) | low (0.0) | back (0.0) |
| ɒ | + + - | velar (0.6) | low (0.0) | back (0.0) |
| æ | + - - | velar (0.6) | low (0.0) | front (1.0) |
| e | + - - | palatal (0.7) | mid (0.5) | front (1.0) |
| ɛ | + - - | palatal (0.7) | mid (0.5) | front (1.0) |
| i | + - + | palatal (0.7) | high (1.0) | front (1.0) |
| ɪ | + - - | palatal (0.7) | high (1.0) | front (1.0) |
| o | + + - | velar (0.6) | mid (0.5) | back (0.0) |
| ɔ | + + + | velar (0.6) | mid (0.5) | back (0.0) |
| ʌ | + - - | velar (0.6) | mid (0.5) | back (0.0) |
| u | + + + | velar (0.6) | high (1.0) | back (0.0) |
| ʊ | + + - | velar (0.6) | high (1.0) | back (0.0) |
| ə | + - - | velar (0.6) | mid (0.5) | mid (0.5) |
| j | - - - | velar (0.6) | high (1.0) | front (1.0) |
| w | - + - | velar (0.6) | high (1.0) | back (0.0) |

Table 4.2: Features for vowels. S, R and L stands for *Syllabic*, *Round* and *Long*, respectively

This score is a bounded integer value. Its boundaries depend on the length of the sequences. In order to know when an extension of a hook exactly matches the keyword, it is necessary to normalize the score values. Given two sequences $a$ and $b$ of lengths $n$ and $m$, with $m$ longer than $n$, the following normalization rule has been used:

$$|\Delta(a,b)| = \frac{\Delta(a,b)}{\frac{\Delta(a,a)}{n} \cdot length(a,b)}$$

where $length(a,b)$ is the length of the best matching between $a$ and $b$. For example, the lengths of both best matches in examples from Figure 4.5 are 8 and 5. The normalized values are between 0 and 1. The final score value $s$ for the extension of a hook $h$ and a keyword $w$ is $|\Delta(w,h)|$.

### Updating Term Frequency

In traditional term-based IR the score is binary. Each term occurrence scores 1. Therefore, the measure of *term frequency* (*tf*) is a particular case of our setting where all the matchings are perfect. We have devised three methods to compute term frequency with non-integer scores. For a given matching score $s$, *tf* can be updated following several strategies:

1. $tf \leftarrow tf + s$

2. $tf \leftarrow \begin{cases} tf + 1 & \text{if } s > t \\ tf & \text{if } s \leq t \end{cases}$

3. $tf \leftarrow \begin{cases} tf + s & \text{if } s > t \\ tf & \text{if } s \leq t \end{cases}$

| Ph. | VNRLT | Manner | Place |
|---|---|---|---|
| b | + - - - - | stop (1.0) | bilabial (1.0) |
| β | + - - - - | fricative (0.8) | bilabial (1.0) |
| c | - - - - - | stop (1.0) | palatal (0.7) |
| d | + - - - - | stop (1.0) | alveolar (0.85) |
| ð | + - - - - | fricative (0.8) | dental (0.9) |
| f | - - - - - | fricative (0.8) | labiodental (0.95) |
| g | + - - - - | stop (1.0) | velar (0.6) |
| h | - - - - - | fricative (0.8) | glottal (0.1) |
| k | - - - - - | stop (1.0) | velar (0.6) |
| l | + - - + - | approx. (0.6) | alveolar (0.85) |
| ʎ | + - - - - | approx. (0.6) | palatal (0.7) |
| m | + + - - - | stop (1.0) | bilabial (1.0) |
| n | + + - - - | stop (1.0) | alveolar (0.85) |
| ŋ | + + - - - | stop (1.0) | velar (0.6) |
| ɲ | + + - - - | approx. (0.6) | palatal (0.7) |
| p | - - - - - | stop (1.0) | bilabial (1.0) |
| r | + - + - + | fricative (0.8) | alveloar (0.85) |
| ɹ | + - + - - | approx. (0.6) | retroflex (0.8) |
| s | - - - - - | fricative (0.8) | alveolar (0.85) |
| ʃ | - - - - - | affricate (0.9) | alveolar (0.85) |
| t | - - - - - | stop (1.0) | alveolar (0.85) |
| θ | - - - - - | fricative (0.8) | dental (0.9) |
| x | - - - - - | fricative (0.8) | velar (0.6) |
| z | + - - - - | fricative (0.8) | alveolar (0.85) |
| ʒ | + - - - - | affricate (0.9) | alveolar (0.85) |

Table 4.3: Features for consonants. V, N, R, L and T stands for *Voice*, *Nasal*, *Retroflex*, *Lateral* and *Trill*, respectively

| Feature | Salience | Feature | Salience |
|---|---:|---|---:|
| Syllabic | 5 | Nasal | 10 |
| Round | 5 | Retroflex | 10 |
| Long | 1 | Lateral | 10 |
| High | 5 | Trill | 10 |
| Back | 5 | Place | 40 |
| Voice | 10 | Manner | 50 |

Table 4.4: Features and their salience.

where $t$ is a fixed threshold.

For example, if two occurrences of a certain word $w$ are found in document $d$ with scores 0.55 and 0.8 respectively, and $t$ is 0.7, term frequency $tf_{w,D}$ is 1.35, 1 and 0.8 respectively for the three methods. The motivation for setting a threshold to filter out some of the matchings is that in some cases the process of similarity detection for a word $w$ will output far more hooks than occurrences of $w$ in the original speech. This specially occurs with words containing very common syllables that will produce a lot of noisy matchings with low similarity score. Initial experiments have shown that the third approach achieves better results. This is why this approach has been used in our experiments.

### 4.4.3 Some Results

We have conducted some experiments of passage retrieval and document retrieval using PHAST. It this experiments we have compared PHAST with other term-detection mechanisms (e.g. full words, character $n$–grams, phone $n$–grams).

For document retrieval we have used PHAST in combination with several classical document ranking models such as Divergence From Randomness [3], Vector Space Models [35] and Okapi BM25 [32]. For passage retrieval we have used the query relaxation algorithm shown in Figure 4.2 (page 23).

We have used a corpus provided by TALP Research Center within the framework of TC-STAR project.[22] It consists of transcripts from European Parliamentary Plenary Sessions mixed with Spanish Parliamentary Session. Manual reference transcript is over 50.000 words long, corresponding to nearly four hours of speech. Automatic transcript has been done by an ASR with an average word error rate of 26.6%. The set of question in Spanish has been written by ourselves.

The results show similar improvement in passage retrieval and in document retrieval on automatic transcripts. Our approach significantly outperforms other systems in 18 points in passage retrieval, and 10 points in document retrieval.

Detailed results can be found in research report [5]. This work has been subtited to ACL 2008 conference.

[22] http://www.talp.upc.edu
http://www.tc-star.org

# Chapter 5

# QAst 2007 Results

This chapter summarizes the QAst 2007 evaluation, the systems that participated and the results achieved, including a detailed analysis of our contribution related to the previous chapter.

## 5.1 Participants

A total of five groups from five different countries submitted results for one or more of the proposed QAst tasks. Due to various reasons (technical, financial, etc.), three other registered groups were not able to submit any results.

The five participating groups are the following:

- CLT, Center for Language Technology, Australia;
- DFKI, Deutsches Forschungszentrum für Künstliche Intelligenz, Germany;
- LIMSI, Laboratoire d'Informatique et de Mécanique des Sciences de l'Ingénieur, France;
- TOKYO, Tokyo Institute of Technology, Japan;
- **UPC**, Universitat Politècnica de Catalunya, Spain.

Five groups participated in both T1 and T2 tasks (CHIL corpus) and three groups participated in both T3 and T4 tasks (AMI corpus).

The participants could submit up to 2 different outputs per task and up to 5 answers per question. The systems used in the submissions are summarized in Table 2.1 in section 2.3. In total, 28 submissions were evaluated: 8 submissions from 5 participating teams for T1, 9 submission files from 5 different sites for T2, 5 submissions from 3 participants for T3 and 6 submissions from 3 participants for T4. The lattices provided for task T2 were not finally used by any participant.

## 5.2 Results

The results for the four QAst tasks are presented in tables 5.1, 5.2, 5.3 and 5.4. Due to some problems (typos, wrong answer types) some questions have been deleted from the scoring results in tasks T1, T2 and T3. Finally, the results have been calculated on the basis of 98 questions for tasks T1 and T2, and 96 for T3. In addition, and due to also missing time information at word level for some AMI meetings, seven questions have been deleted from the scoring results of T4. The results for this task have been calculated on the basis of 93 questions.

| System | Questions | Correct answers | MRR | Accuracy |
|---|---|---|---|---|
| clt1_t1 | 98 | 16 | 0.09 | 0.06 |
| clt2_t1 | 98 | 16 | 0.09 | 0.05 |
| dfki1_t1 | 98 | 19 | 0.17 | 0.15 |
| limsi1_t1 | 98 | 43 | 0.37 | 0.32 |
| limsi2_t1 | 98 | 56 | 0.46 | 0.39 |
| tokyo1_t1 | 98 | 32 | 0.19 | 0.14 |
| tokyo2_t1 | 98 | 34 | 0.20 | 0.14 |
| upc1_t1 | 98 | 54 | 0.53 | 0.51 |

Table 5.1: Results for T1 (QA on CHIL manual transcriptions)

| System | Questions | Correct answers | MRR | Accuracy |
|---|---|---|---|---|
| clt1_t2 | 98 | 13 | 0.06 | 0.03 |
| clt2_t2 | 98 | 12 | 0.05 | 0.02 |
| dfki1_t2 | 98 | 9 | 0.09 | 0.09 |
| limsi1_t2 | 98 | 28 | 0.23 | 0.20 |
| limsi2_t2 | 98 | 28 | 0.24 | 0.21 |
| tokyo1_t2 | 98 | 17 | 0.12 | 0.08 |
| tokyo2_t2 | 98 | 18 | 0.12 | 0.08 |
| upc1_t2 | 96 | 37 | 0.37 | 0.36 |
| upc2_t2 | 97 | 29 | 0.25 | 0.24 |

Table 5.2: Results for T2 (QA on CHIL automatic transcriptions)

The results are very encouraging. First, the best result in accuracy achieved in tasks involving manual transcripts (0.51 for task T1) is closed to the best two results for factual questions in TREC 2006 (0.58 and 0.54), in which monolingual English QA was evaluated. Second, this behavior is also observed in average: the accuracy in average achieved in tasks T1 and T3 is 0.22, which is comparable with 0.18 achieved in TREC 2006. Although no direct comparisons between QAst and TREC are possible due to the use of different data, questions and answer types, these facts show that QA technology can be useful to deal with spontaneous speech transcripts.

Finally, the accuracy values are 0.22 and 0.15 in average for the tasks involving lectures (T1 and T2, respectively), and 0.21 and 0.14 for those involving meetings (T3 and T4, respectively). These values show that the accuracy decreases in average more than 36% when dealing with automatic transcripts. The reduction of this difference between accuracy values have to be

[23] Due to a bug with the output format script, we asked to the assessors to reevaluate our unique run for T3. The results in brackets must be regarded as a non official run.

| System | Questions | Correct answers | MRR | Accuracy |
|---|---|---|---|---|
| clt1_t3 | 96 | 31 | 0.23 | 0.16 |
| clt2_t3 | 96 | 29 | 0.25 | 0.20 |
| limsi1_t3 | 96 | 31 | 0.28 | 0.25 |
| limsi2_t3 | 96 | 40 | 0.31 | 0.25 |
| upc1_t3<br>post evaluation[23] | 95<br>95 | 23<br>27 | 0.22<br>0.26 | 0.20<br>0.25 |

Table 5.3: Results for T3 (QA on AMI manual transcriptions).

| System | Questions | Correct answers | MRR | Accuracy |
|---|---|---|---|---|
| clt1_t4 | 93 | 17 | 0.10 | 0.06 |
| clt2_t4 | 93 | 19 | 0.13 | 0.08 |
| limsi1_t4 | 93 | 21 | 0.19 | 0.18 |
| limsi2_t4 | 93 | 21 | 0.19 | 0.17 |
| upc1_t4 | 91 | 22 | 0.22 | 0.21 |
| upc2_t4 | 92 | 17 | 0.15 | 0.13 |

Table 5.4: Results for T4 (QA on AMI manual transcriptions)

taken as a main goal in the future research.

## 5.3 About Our Results

After the QAst official evaluation we discovered and corrected some bugs in our system. It helped improving the scores. The present section contains a detailed analysis of our results taking into account this updated scores, which differ from the official evaluation in Section 5. Table 5.5 summarizes our new overall results.

In the tasks based on manual transcripts (T1 and T3, Tables 5.1 and 5.3) we submitted one run using the system described in Section 4.2.1. We refer to this system as QA$_m$. In the tasks based on automatic transcripts (T2 and T4 Tables 5.2 and 5.4) we submitted two runs: one using QA$_m$ and another using the system tailored for automatic transcripts, where we used the PHAST keyword matching algorithm (see Section 4.4) and the NERC expanded with phonetic attributes (Section 4.3.2). We refer to the latter system as QA$_a$.

| Task and System | #Question | MRR | Accuracy | TOP1 | TOP5 |
|---|---|---|---|---|---|
| T1, QA$_m$ | 98 | 0.53 | 0.51 | 50 | 54 |
| T2, QA$_a$ | 98 | 0.34 | 0.34 | 33 | 36 |
| T2, QA$_m$ | 98 | 0.36 | 0.36 | 35 | 36 |
| T3, QA$_m$ | 96 | 0.37 | 0.34 | 34 | 42 |
| T4, QA$_a$ | 93 | 0.17 | 0.17 | 16 | 19 |
| T4, QA$_m$ | 93 | 0.22 | 0.21 | 20 | 22 |

Table 5.5: Overall results for the four QAst tasks. Some bugs where fixes after the evaluation

A first glimpse at the scores in Table 5.5 indicates that the results obtained are 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 5.5 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 0.12 in the AMI corpus. In relative terms, this is a drop of the TOP1 score of 29% in CHIL and 35% 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. The performance drop is smaller for the AMI corpus than CHIL corpus. The explanation is that, AMI transcripts had a higher Word Error Rate (WER) than the CHIL transcripts (38% versus 20%) and 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 follows WER in small corpora with little redundancy (e.g., CHIL) and is smaller than 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 deal with automatic transcripts. The explanation is twofold. 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.

### 5.3.1 Error Analysis

Table 5.6 shows the distribution of correct answers according to the answer type for all tasks. The table indicates that our system had a particularly hard time answering questions in task T3/T4, when the answer type was a NE of types Org, Loc, Tim, or Mea. 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 could be to train an open-domain NERC, where large annotated

| 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 | - | - | - |
| T2, $QA_a$ | 6/20 | 4/9 | 2/9 | 6/10 | 10/28 | 5/18 | 3/4 | - | - | - |
| T2, $QA_m$ | 8/20 | 3/9 | 3/9 | 6/10 | 7/28 | 7/18 | 2/4 | - | - | - |
| T3, $QA_m$ | 5/13 | 8/15 | 6/14 | 1/14 | 4/12 | - | 1/2 | 5/9 | 4/6 | 8/11 |
| T4, $QA_a$ | 2/13 | 3/15 | 2/14 | 1/14 | 2/12 | - | 0/2 | 3/9 | 1/6 | 4/11 |
| T4, $QA_m$ | 3/13 | 2/15 | 3/14 | 1/14 | 4/12 | - | 1/2 | 3/9 | 1/6 | 5/11 |

Table 5.6: Distribution of correct answers (TOP5) according to answer type. Org = organization, Per = person, Tim = time, Mea = measure, Met/Sys = method/system, Mat = material, Col = color

corpora are available, and use domain transfer techniques to adapt the open-domain system to the AMI domain. The performance drop-off between manual and automatic transcripts is similar in all types of NE.

| Task and System | Questions | QC<br>Correct | PR<br>Correct | QC & PR<br>Correct | TOP1 | TOP5 |
|---|---|---|---|---|---|---|
| T1, $QA_m$ | 89 | 62 | 74 | 53 | 47 | 50 |
| T2, $QA_a$ | 89 | 60 | 62 | 41 | 29 | 32 |
| T2, $QA_m$ | 89 | 60 | 61 | 43 | 32 | 33 |
| T3, $QA_m$ | 90 | 80 | 68 | 60 | 28 | 34 |
| T4, $QA_a$ | 90 | 80 | 50 | 46 | 12 | 13 |
| T4, $QA_m$ | 90 | 80 | 52 | 49 | 16 | 19 |

Table 5.7: Error analysis of the QA system components for non-nil question

Table 5.7 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 which answer is not "NIL". 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. This table only considers questions with a non–null answer. As it can be seen, answer extraction is much more difficult in tasks T3 and T4. Less than a half of the possible answers (correct PR and QC) are ultimately extracted in the first place. It shows that T3 and T4 are specially difficult for NERC or AE modules. Passage Retrieval is also more difficult in T3 and T4 tasks but this difference is much smaller.

misclassified, misclassified, misclassified, misclassified, misclassified, misclassified, misclassified, misclassified, misclassified,

Table 5.8 summarizes the joint error analysis of NERC, PR and AE. The "PR Correct" column shows the number of questions where at least one passage with the correct answer was retrieved. The "NE tag Correct" columns shows the number of questions with correct

| Task and System | Rankings | | Candidate Answers | | NERC Errors | | |
|---|---|---|---|---|---|---|---|
| | Top5 | Top 1 | PR Correct | NE tag Correct | No NE | Misclassified | Other Error |
| T1, QA$_m$ | 50 | 47 | 74 | 39 | 10 | 22 | 11 |
| T2, QA$_a$ | 32 | 29 | 62 | 28 | 44 | 18 | 8 |
| T2, QA$_m$ | 33 | 32 | 61 | 28 | 22 | 10 | 9 |
| T3, QA$_m$ | 34 | 28 | 69 | 39 | 14 | 12 | 7 |
| T4, QA$_a$ | 13 | 12 | 50 | 21 | 23 | 3 | 7 |
| T4, QA$_m$ | 19 | 16 | 52 | 22 | 26 | 3 | 6 |

Table 5.8: Error analysis of the NERC and PR components

passages containing the answer correctly tagged by the NERC as the same type than QC; it is an upper bound for AE step. The "NERC Errors" columns decompose the NERC errors in three categories. It shows that the most of the errors are untagged NEs (the error count is over the whole set of correct passages retrieved in PR, sometimes more than one per question). For tasks T1 and T2, TOP1 score i bigger than the upper bound. This is due to a fall-back mechanism implemented in the AE step that can overcome some usual question classification errors (e.g. person instead of organization or location). The numbers show how NE tagging is the major source of errors while answer extraction (difference between TOP1 and correct NE tag) is far more robust than it.

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.

- 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 and the conclusion was that in most of the cases the fault can be assigned to the NERC prediction, 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 [40]. Therefore, NERC for automatic transcripts needs a big improvement.

## 5.4 QAst Conclusions

### 5.4.1 QAst General Results

The QAst results achieved show that, first, QA technology can be useful to deal with spontaneous speech transcripts, and second, the loss in accuracy when dealing with automatically transcribed speech is high.

A set of five groups participated in QAst with a total of 28 submitted runs among four specific tasks. We were one of the few participants that submitted runs in all four QAst 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).

### 5.4.2 Our Participation in QAst

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., it 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 work 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. NERC enhanced with phonetic features deserves further research.

# Chapter 6

# PhD. Thesis Project

*—I did the right thing, didn't I? It all worked out in the end.*
*—In the end? Nothing ends, Adrian. Nothing ever ends.*

*Watchmen* - Alan Moore

In this document we have presented an approach to QA on automatic speech trancripts. We have developed an specific QA system for this scenario (Chapter 4) and have pointed out its main drawbacks and possible solutions. From now on, we will continue the resarch in the three principal branches presented here, e.g. Information Retrieval, Named Entity Recognition and Answer Extraction.

## 6.1 Short Term Work

Our most immediate work will be the organization of QAst track in CLEF 2008. In this edition the evaluation will be enhanced with the following features:

- The task will cease to be monolingual (as it is the aim of CLEF). This year there will be corpora for three language: English, French and Spanish.
- QAst participants will be provided with 3 different versions of automatic transcripts corresponding to the outputs of 3 different ASR systems with 3 different WER values. Also, they will be provided with the lattices of the best output. Then will be possible to evaluate the impact of WER in performance.

The track starts with the release of the development data set in February and the final submission deadline will be in July.

We will continue our research in robust algorithms for SDR. We plan to enrich our methodology with error-detection mechanisms. Works such as [37, 14] show that the use of semantic information helps to discriminate incorrect words in the ASR output. This can be used in combination with phonetic similarity to reduce the excess of noise that our current methodology introduces and provide more accurated retrieval.

As we have shown in Section 5.3, our NERC module needs to improve. We plan to enrich our machine learning models with phonetic information and error-detection mechanisms similar to passage retrieval.

## 6.2 Mid Term Work

Using new mechanisms to add robustness to Answer Extraction. Given that NERC may become a bottle-neck in our research, we plan to use Semantic Roles to add robustness to Answer Extraction. A semantic role is the relationship that a syntactic constituent has with a predicate. Typical semantic arguments include Agent, Patient, Instrument, etc. and also adjunctive arguments indicating Locative, Temporal, Manner, Cause, etc. aspects. Semantic Role Labeling (SRL) is a key task for answering "Who", "When", "What", "Where", etc. questions. Although SRL is a task from a higher abstraction level than NERC, we expect that this will be more robust to automatic speech recognition than NERC. Even though this subject is not directly related to QA, we have recent works in the field of SRL. We hope this works can also be successfully applied to spoken language. Our works on SRL [25, 24, 41] are reported in Chapter 7.

We will also organize the future 2009 QAst evaluation, where we expect to participate with most of the improvements mentioned here.

## 6.3 Work Plan

It is expected to conclude the PhD thesis in two years from now, in spring of 2010. Experiments with the research lines presented should take a year and a half, and then the doctoral thesis is expected to be finished in the following 6 months.

We will devote the rest of 2008 to research on Information Retrieval and NERC subjects as well as the immediate organization of QAst 2008. In year 2009 we plan to work mainly in AE and the use of SRL in automatic transcripts. We will also organize QAst 2009. The writing and revision of the doctoral thesis dissertation will start probably after completion of QAst evaluation in summer 2009.

Figure 6.1 shows the devised scheduling.

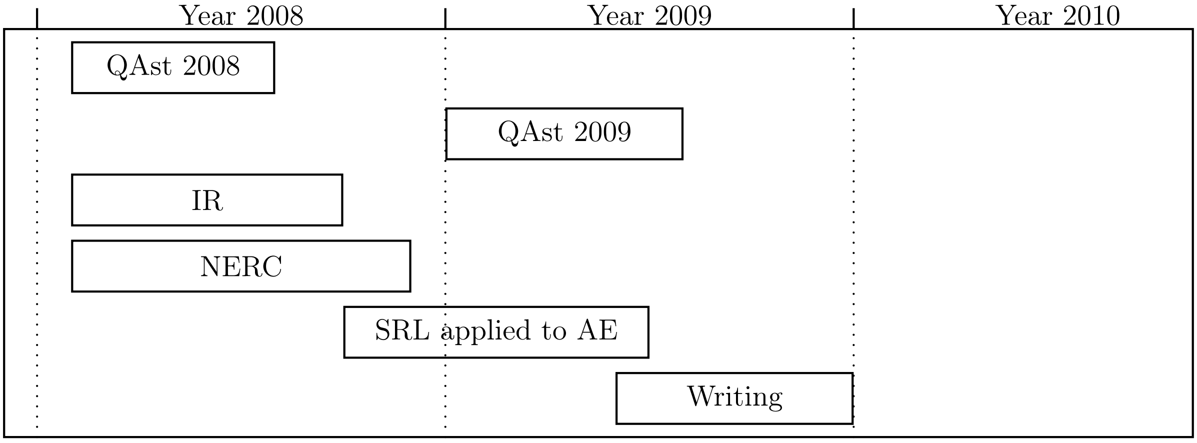

Figure 6.1: PhD. Thesis Scheduling

# Chapter 7

# Publications

*"El amontonamiento de miles y millones de referencias en una obra literaria; los desfiles etimológicos, fraseológicos, hermenéuticos; la superposición de sentidos interminables y maliciosamente antinómicos, no es una creación artística, sino la elaboración de pasatiempos intelectuales para tipos particularmente paranoicos, para maníacos y coleccionistas que buscan la excitación en el manejo de las bibliografias."*

*Vacío Perfecto* - Stanisław Lem

This section summarizes the publications describing our research in fields related to Question Answering.

## 7.1 Question Answering

- [Surdeanu and Dominguez and Comas]
  Mihai Surdeanu, David Dominguez-Sal and Pere R. Comas.
  Design and Performance Analysis of a Factoid Question Answering System for Spontaneous Speech Transcriptions
  Proceedings of the Ninth International Conference on Spoken Language Processing (INTERSPEECH 2006), ICSLP. Pittsburg, USA. September, 2006.

  *QA system tailored for manual transcripts of spontaneous speech. It uses only robust NLP components.*

- [Comas and Turmo and Surdeanu, 2007]
  Pere R. Comas, Jordi Turmo and Mihai Surdeanu.
  Robust Question Answering for Speech Transcripts Using Minimal Syntactic Analysis
  Cross Language Evaluation Forum (CLEF). September, 2007.

*This is the UPC proposal to QAst 2007 task. Its content is explained in Chapter 4.*

- [Turmo et al., 2007]
  Jordi Turmo, Pere R. Comas, Christelle Ayache, Djamel Mostefa, Sophie Rosset and Lori Lamel.
  Overview of QAST 2007
  Cross Language Evaluation Forum (CLEF). September, 2007.

  *Results overview of QAst 2007. Its content is expanded in Chapters 3 and 5.*

- [Lamel et al., 2007]
  Lori Lamel, Sophie Rosset, Christelle Ayache, Djamel Mostefa, Jordi Turmo and Pere R. Comas
  Question Answering on Speech Transcriptions: the QAST evaluation in CLEF
  International Conference on Language Resources and Evaluation (LREC). May 2008.

  *This aricle if for dissemination of QAst research results in LREC conference.*

## 7.2 Spoken Document Retrieval

- [Comas and Turmo, 2008]
  Pere R. Comas and Jordi Turmo.
  PHAST: Spoken Document Retrieval Based on Sequence Alignment
  Report de Recerca del LSI: LSI-08-2-R

  *Contains a complete description of PHAST algorithm and reports experimental results on SDR oriented to QA.*

## 7.3 Semantic Role Labeling

Research in Semantic Role Labeling started in 2005 as a paralel work to the Thesis and hence this publications have not been discussed in detail. As stated in Chapter 6, we plan to join both research lines in the immediate future.

- [ Màrquez et al. 2005a]
  Lluís Màrquez, Pere R. Comas, Jesús Giménez and Neus Català
  Semantic Role Labeling as Sequential Tagging
  Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL'05), pg. 193–196. 2005.

  *This is the UPC proposal to CoNLL 2005 shared task on SRL. Reduces SRL problem to a sequential tagging using AdaBoost. It ranked third.*[24]

- [Màrquez et al. 2005b]
  Lluís Màrquez, Mihai Surdeanu, Pere R. Comas and Jordi Turmo.
  A Robust Combination Strategy for Semantic Role Labeling
  Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'05), pg. 644–651. October, 2005.

[24] `http://www.lsi.upc.edu/~srlconll/`

*Description of a machine-learning strategy to combine the output of three different SRL systems.*

- [Surdeanu et al. 2007]
  Mihai Surdeanu, Lluís Màrquez, Xavier Carreras and Pere R. Comas.
  Combination Strategies for Semantic Role Labeling
  Journal of Artificial Intelligence Research, 29, pg. 105–151. June, 2007.

  *Extensive experiments with SRL combination techniques. It uses several machine-learning systems and optimization algorithms.*

# Acknowledgements

This research has been partially supported by the European Commission (CHIL project, IP-506909; PASCAL Network, IST-2002-506778) and the Spanish Ministry of Education and Science (TRANGRAM, TIN2004-07925-C03-02).

# Bibliography

[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] M. Alzghool and D. Inkpen. University of Ottawa's participation in the CL-SR task at CLEF 2006. *In Proceedings of the CLEF 2006 Workshop on Cross-Language Information Retrieval and Evaluation*, 2006.

[3] G. Amati and C.J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389, 2002.

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

[5] P.R. Comas and J. Turmo. Phast: Spoken document retrieval based on sequence alignment. *Report de Recerca del LSI: LSI-08-2-R*, 2008.

[6] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research - MIT Press*, 2003.

[7] D. Ferrés. Geographical information resolution and its application to the question answering systems. *Memòria del DEA i Projecte de Tesi. Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya*, 2006.

[8] J. Garofolo, G. Auzanne, and E. Voorhees. The TREC spoken document retrieval track: A success story. *Proceedings of the Recherche d'Informations Assiste par Ordinateur: ContentBased Multimedia Information Access Conference*, 2000.

[9] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan. The ami system for the transcription of meetings. *Proceedings of ICASSP'07*, 2007.

[10] S. Harabagiu, D. Moldovan, and J. Picone. Open-domain voice-activated question answering. *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*, 2002.

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

[12] D. Inkpen, M. Alzghool, and A. Islam. Using various indexing schemes and multiple translations in the CL-SR task at CLEF 2005. *In Proceedings of CLEF 2005, Lecture Notes in Computer Science 4022, Springer-Verlag*, 2006.

[13] D. Inkpen, M. Alzghool, G. Jones, and D.W. Oard. Investigating cross-language speech retrieval for a spontaneous conversational speech collection. In *HLT-NAACL*, 2006.

[14] D. Inkpen and A. Désilets. Semantic similarity for detecting recognition errors in automatic speech transcripts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*, 2005.

[15] G.J.F. Jones, K. Zhang, and A.M. Lam-Adesina. Dublin city university at CLEF 2006: Cross-language speech retrieval (CL-SR) experiments. *In Proceedings of the CLEF 2006 Workshop on Cross-Language Information Retrieval and Evaluation*, 2006.

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

[17] G. Kondrak. A new algorithm for the alignment of phonetic sequences. *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 288–295, 2000.

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

[19] L. Lamel, G. Adda, E. Bilinski, and J.-L. Gauvain. Transcribing lectures and seminars. *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2005)*, 2005.

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

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

[22] D. Mollá, S. Cassidy, and M. van Zaanen. Answerfinder at QAst 2007: Named entity recognition for qa on speech transcripts. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.

[23] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. Chu, A. Tyagi, J Casas, J. Turmo, L. Cristoforetti, F. Tobia, A. Pnvmatikakis, V. Mylonakis, F. Talantzis, S. Burger, R. Stiefelhagen, L Bernardin, and C. Rochet. The chil audiovisual corpus for lecture and meeting analysis inside smart rooms. *to appear in Language Resources and Evaluation Journal*, 2007.

[24] L. Màrquez, P.R. Comas, J. Giménez, and N. Català. Semantic role labeling as sequential tagging. *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL'05)*, 2005.

[25] L. Màrquez, M. Surdeanu, P.R. Comas, and J. Turmo. A robust combination strategy for semantic role labeling. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*, 2005.

[26] G. Neumann and R. Wang. DFKI-LT at QAST 2007: Adapting QA components to mine answers in speech transcripts. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.

[27] D.W. Oard, J. Wang, G.J.F. Jones, R.W. White, P. Pecina, D. Soergel, X. Huang, and I. Shafran. Overview of the CLEF-2006 cross-language speech retrieval track. *Proceedings of the CLEF 2006 Workshop on Cross-Language Information Retrieval and Evaluation*, 2006.

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

[29] P. Pecina, P. Hoffmannová, G.J.F. Jones, Y. Zhang, and D. Oard. Overview of the CLEF-2007 cross-canguage speech retrieval track. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.

[30] C. Peters, P. Clough, F.C. Gey, J. Karlgren, B. M;agnini, D.W. Oard, M. de Rijke, and M. Stempfhuber, editors. *Evaluation of Multilingual and Multi-modal Information Retrieval.* Springer-Verlag., 2006.

[31] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[32] S.E. Robertson, S. Walker, K. Spärck-Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D.K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3).* Gaithersburd, MD: NIST, 1995.

[33] S. Rosset, O. Galibert, G. Adda, and E. Bilinski. The LIMSI participation in the QAst track. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.

[34] G. Salton, editor. *Automatic text processing.* Addison-Wesley Longman Publishing Co., Inc., 1988.

[35] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Cornell University, 1987.

[36] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *HLT-NAACL*, 2004.

[37] A. Sarma and D. Palmer. Context-based speech recognition error detection and correction. *HLT-NAACL*, 2004.

[38] S. Srinivasan and D. Petkovic. Phonetic confusion matrix based spoken document retrieval. In *SIGIR*, 2000.

[39] S. Stenchikova, D. Hakkani-Tür, and G. Tur. Qasr: Question answering using semantic roles for speech interface. *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2006)*, 2006.

[40] 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.

[41] M. Surdeanu, L. Màrquez, X. Carreras, and P.R. Comas. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research (JAIR)*, 2007.

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

[43] J. Turmo, P.R. Comas, C. Ayache, D. Mostefa, S. Rosset, and L. Lamel. Overview of QAST 2007. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.

[44] B. van Schooten, S. Rosset, O. Galibert, A. Max, R. op den Akker, and G. Illouz. Handling speech input in the ritel qa dialogue system. *Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2007)*, 2007.

[45] E.M. Voorhees and L.L. Buckland, editors. *The Fifteenth Text Retrieval Conference (TREC 2006) Proceedings*, 2006.

[46] J. Wang and D.W. Oard. CLEF-2005 CL-SR at maryland: Document and query expansion using side collections and thesauri. *In Proceedings of the CLEF 2005 Workshop on Cross-Language Information Retrieval and Evaluation*, 2005.

[47] R.W. White, D.W. Oard, G.J.F. Jones, D. Soergel, and X. Huang. Overview of the CLEF-2005 cross-language speech retrieval track. *Proceedings of the CLEF 2005 Workshop on Cross-Language Information Retrieval and Evaluation*, 2005.

[48] E.W.D. Whittaker, J. Novak, P. Chatain, and S. Furui. TREC 2006 question answering experiments at tokyo institute of technology. *The Fifteenth Text Retrieval Conference (TREC 2006) Proceedings*, 2006.

[49] E.W.D. Whittaker, J.R. Novak, M. Heie, and S. Furui. CLEF2007 question answering experiments at tokyo institute of technology. *Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation*, 2007.