Ph.D. Thesis
This Thesis is presented for the degree of Doctor of Philosophy

Factoid Question Answering for Spoken Documents

by
Pere R. Comas i Umbert

advised by
Dr. Jordi Turmo
Dr. Lluís Màrquez

Ph.D. Programme in Artificial Intelligence
Departament de Llenguatges i Sistemes Informàtics (LSI)
Universitat Politècnica de Catalunya (UPC)

Barcelona, April 2012

Dissertation advisor:

Dr. Jordi Turmo i Borràs
Departament de Llenguatges i Sistemes Informàtics
Facultat d'Informàtica de Barcelona
Universitat Politèctica de Catalunya

Dissertation co-advisor:

Dr. Lluís Màrquez i Villodre
Departament de Llenguatges i Sistemes Informàtics
Facultat d'Informàtica de Barcelona
Universitat Politèctica de Catalunya

Examination Committee:

Dr. Lluís Padró, Universitat Politècnica de Catalunya, Spain

Prof. Dr. Maarten De Rijke, University of Amseterdam, The Netherlands

Dr. Horacio Rodríguez, Universitat Politècnica de Catalunya, Spain (chair)

Dr. Sophie Rosset, LIMSI-CNRS, France

Dr. Jose Luís Vicedo, Universidad de Alicante, Spain

Date of public defense: June 12th, 2012

Acknowledgments

The work leading to this thesis has been partially funded by the European Community's Sixth Framework Programme CHIL project (Computer in the Human Interaction Loop), IST-2004-506909, and by the Spanish Ministry of Science and Innovation projects TEXT-MESS (TIN2006-15265-C06), and KNOW-2 (TIN2009-14715-C04-04).

Table of Contents

1. Introduction

1.1 Motivation

Radio, television and social video sites provide immediate access to an ever-growing amount of multimedia documents containing potentially useful information. Nowadays multimedia content, especially video and audio recordings, is used to convey a large variety of information. Although some of these spoken documents may have a written counterpart elsewhere (i.e., broadcast news), most of their valuable information can only be found in spoken form. Videos from on-line university courses, pod-casts, recorded meetings and presentations are obvious examples of this. The content of these spoken documents may be sorted, classified, described and tagged in text form, but ultimately they must be accessed by either listening to the audio or reading written transcripts, where available.

Accessing large collections of audio files is difficult and intrinsically time-consuming. Browsing audio recordings is a difficult task for humans, and reviewing such documents quickly is technically difficult [Fayolle et al., 2010]. Automatic speech recognition (ASR) produces searchable text but is also expensive in terms of both time and computational resources. ASRs are designed for very concrete recording conditions and/or speakers and are not generally applicable to every spoken document. Hence, speaker independent large vocabulary continuous ASRs are not particularly reliable. ASR transcripts contain highly noisy data because the word recognition errors affect significant content words, especially named entities (e.g., person names, organisation names, locations…), and the lack of capitalisation and punctuation. Although it is possible to reconstruct the case and add punctuation to the transcripts [Paulik et al., 2008, Batista et al., 2008, Fayolle et al., 2010], the quality is far from perfect. For example, Batista et al. [2008] achieves an F_1=0.82 for capitalisation and F_1=0.70 for full-stop detection tasks on automatic transcripts of broadcast news. Producing human-made transcripts for all these spoken documents it could be an alternative to ASR, but it is slow, expensive, and probably infeasible.

Spoken document retrieval (SDR) is the application of information retrieval (IR) techniques to retrieve spoken documents from a collection. SDR works with transcripts of the spoken documents (usually ASR transcripts), and uses standard IR algorithms and techniques [Pecina et al., 2007]. Thus, the output of an SDR engine is a list of spoken documents sorted according their relevance to a set of query terms given by the user. After this process of information retrieval, the user has to peruse the documents, or segments of them, in search of anything of his interest.

Question Answering (QA) is the task of extracting short, relevant textual answers in response to natural language questions. QA may use IR techniques, but it is different from IR as it outputs concrete answers to a question instead of references to full documents that are relevant to a query. QA systems are usually classified according to what type of questions they can answer: *factoid* questions are those whose answers are semantic entities (e.g., organisation names, person names, dates, etc.). For example, the question "*Which country is the city of Fallujah in?*" is factoid and the answer, *Iraq*, can be found in relevant documents. In opposition to factoid questions, *definitional* ones ask for interesting information about a topic or entity. For example, "*What is the Ombudsman's office?*" is a definitional question. *List questions* ask for different instances of a particular kind of information to be returned [Wang et al., 2008]. "*What are the different political groupings within the European Parliament?*" is a list question that requires the names of political parties. Finally, the *complex* question class [Harabagiu et al., 2006] is frequently used when the question refers to relations between entities and events, or scenarios involving deep knowledge of the topic, as in *why* [Verberne et al., 2007], and *how* questions [Weber et al., 2012]. Systems oriented to work with *opinions* rather than facts can also be included in this category [Dang and Owczarzak, 2008]. Some of these complex question systems also use automatic-summarisation techniques to produce the answers [Chali et al., 2009].

QA systems may have additional the capacity to answer a series of questions which are anaphorically related to a topic (also known as *context questions*), or to engage in interactive use with humans in a dialogue-style interaction [Dang et al., 2007]. Voice interfaces for QA systems have received the attention of researchers in recent years [Harabagiu et al., 2002]. These systems (e.g., spoken question answering services for limited display devices such as mobile phones) receive the name of *voice activated QA* or *spoken QA* systems. Spoken QA systems focus on integrating QA and ASR technology in one pipeline. In some cases, interactive spoken QA systems may take advantage of the interactivity of the scenario to refine the transcription and question analysis to achieve better performance.

Orthogonally to this system classification based on question types, we can also classify *restricted domain* and *open domain* QA systems. The difference between these classes is that the former exploit domain-dependent resources, such as terminology dictionaries and restricted-domain ontologies [Mollá and Vicedo, 2007]. In recent years, extensive evaluation resources for QA have been created for different domains and information sources [Voorhees, 2004, Peñas et al., 2010].

In addition to standard written documents, it is natural to extend QA research to audio and video media. Furthermore, it is natural for a speech retrieval interface to provide the most accurate clues to users, as it is much more difficult for a human to browse through long audio records than through written records. Thus, giving a concrete answer (i.e., a short segment of speech) to a user query is very important when working with spoken documents. This is the main motivation of QA from spoken documents. Current QA systems use natural language technology that requires text written in accordance with standard rules for written grammar. However, the "grammar" of spoken language is quite different from that of written language. Speech contains disfluencies, repetitions, restarts and corrections. Moreover, any practical application of a search within speech requires the transcripts to be produced automatically with ASRs, which introduces a number of errors. For these reasons, almost any QA system for spoken documents works basically as a pipeline of an ASR system and a regular QA system, and uses only very shallow linguistic processing (i.e., part-of-speech tagging and named entity recognition), as this is more robust than complex analysers like syntactic parsers or coreference resolution systems.

1.2 The Nature of Spoken Documents

The combination of transcript errors and loose discourse and syntax are the most immediate impediments when working with spoken documents. The following examples illustrate some of the issues that must be taken into account when designing a spoken document QA system:

- If the speech corresponds to a multi-part conversation (e.g. telephone conversations, meetings, debates), the discourse structure will be significantly more complex than that of a monologue (e.g. lectures, speeches, broadcast news). In the former, it is crucial to have an automatic speaker detection for consistent interpretation of the discourse, and the turn structure (e.g. dialogue, debate) should be specifically addressed in the QA.

Manual Transcript

B: Uh right, so you want an animal and the characteristics of that animal. Do you have to be able to recognise what animal it is? Um Only animal I could thin- I could draw.
A: Uh I do not think so, I think it's just to try out the whiteboard. Ah.
C: Are we all gonna draw a cat?
D: I know.
B: Its a sort of bunny rabbit cat. You can tell it's not a bunny rabbit by the ears. Um I suppose it should have a mouth as well, sort of.
…

Extracted from the AMI corpus of meetings

- Recordings in noisy scenarios or with far-distance microphones will probably contain much more ASR recognition errors, as in the previous example, than clean, close-distance ones. The sample speech on the right is taken from the European Parliament Plenary Sessions corpus. It has a clear syntax and a close-distance recording in a (quite) silent environment. We present it with three different automatic transcripts obtained from three ASRs with different characteristics. ASR **A** is the most precise of the three, while ASR **C** is the least, as it will be detailed in Section 2.3.4. The automatic recognition is good in all transcripts, although all ASRs have problems with the word “*endure*”and with “*hand*,” which may sound much similary to “*and*.”

Manual	**ASR A**	**ASR B**	**ASR C**
this	this	this	this
really	really	really	really
is	is	is	is
the	the	the	the
last	last	last	last
afternoon	afternoon	afternoon	often
that	that	that	in
the	the	the	the
House	House	House	house
will	will	will	will
have	have	have	have
to	to	to	to
endure me	in Germany	in Germany	endure a make
but	-	the	but
that is	that's	debts	that
of	of	of	of
course	course	course	course
in	in	in	in
the	the	the	the
Parliament's	Parliament	Parliament	parliament
hands	and	and	and
(hesitation)	and	I am	and
as	as	as	as
Honourable	honourable	honourable	honorable
Members	Members	Members	members
know	know	know	know
given	given	given	given
the	the	the	the
way	way	way	way
in	in	in	in
which	which	which	which
the	the	the	the
…	…	…	…

Sample from the EPPS corpus

- Many QA systems are tailored for a specific domain and use domain-dependent information, e.g. specific ontologies, gazetteers [Mollá and Vicedo, 2007]. For QA on spoken documents, it is also important that the system can be adapted to the document's kind of orality (e.g. read text, planned speech, spontaneous speech). For example, read text is closer to written text than spontaneous speech. The latter contains many more disfluencies and grammatical inconsistencies from the viewpoint of standard grammars, thus becoming less suitable for text-based natural language processing tools. On the side there is an example of spontaneous speech taken from a seminar lecture.

Manual Transcript	**ASR Transcript**
we worked on a	we worked on
in a multilingual way	which meeting will play
and we presented at	and be presented at
Eurospeech two thou-	EUROSPEECH two
sand and three ,	thousand three
together with Stefan	toghether with different
Kantak who then also	kind with and also pre-
presented a paper on	sented a paper on multi-
multilingual	lingual
uh grapheme based	grapheme based speech
speech recognition .	recognition
but we had the advan-	but we have the ad-
tage that we had the	vantage that we have
globalphone corpus	Global Phone corpus
which could	which could
gave us better compara-	give us better compare
ble results ,	results

Sample extracted from the TED corpus

- An ASR can only recognise a finite set of words. These are the words belonging to the ASR's *language model*, which estimates the probability of a certain sequence of words. This yield a problem of unrecognisable out of vocabulary words (OOV). Additionally, ASRs have recognition errors also for in-vocabulary words. Section 5.2.1 provides further explanations on this topic. The example on the right shows an utterance with OOV words on the ASR transcript.

ASR Transcript	Manual Transcript
thank	Thank
you	you
i	Hi
good	good
afternoon	afternoon
–	uh
i messages	I'm Yasushi Ishikawa with
fell the into selected	Mitsubishi Electric
corporation	Corporation
and	and
at	uh
i would	I'd

Sample extracted from the TED corpus

1.3 Objectives and Contributions of this Thesis

The main objective of our work has been to design, implement and evaluate a QA system to specifically deal with spoken documents, creating and evaluating new techniques whenever necessary to address specific issues related to spoken documents. We have called our QA system Sibyl after the ancient Greek oracle. The contributions resulting of this work can be grouped into these two categories:

- The Sibyl question answering system: This is a fully functional QA system developed in this thesis whose main features are its robustness and easy portability to new domains or kinds of orality. From this perspective, Sibyl does not rely on hand-crafted knowledge and it is language-independent. Sibyl uses several linguistic analysers and automatically learnt models in its pipeline. Given the suitable set of examples, it is possible to learn new models for other languages and/or scenarios with little human supervision. Sibyl has been tested within the QAst framework (see Section 2.3) for different scenarios and ASRs, achieving state-of-the-art results compared to knowledge-based QA systems. The results show that syntax can be used to improve the performance of QA on speech transcripts even for automatic recognition. Sibyl can use coreference resolution, but our experiments show that it has minimal impact in the spoken document collections we have used.

- The creation of the first evaluation framework for the task of question answering on spoken documents: This framework, called QAst, was introduced as a pilot track in the CLEF[1] workshop of 2007 and lasted for 3 years. Each year different data and

[1] http://www.clef-initiative.eu

evaluation scenarios have been released. This evaluation framework has helped the creation of literature on question answering on spoken documents and to settle this as a stand-alone research topic.

1.4 Overview of this Document

This dissertation is organised as follows:

- Chapter 2 explains the usual architecture of a QA system, reviews the state-of-the-art of factoid QA on spoken documents and provides complete details about the QAst evaluation framework.
- Chapter 3 is a short overview of SIBYL. It shows the architecture of the system and how its parts are related among them. The next Chapters individually describe the main modules of SIBYL and how they tackle spoken documents. Evaluation results for each part are detailed in each chapter.
- Chapter 4 describes how the questions are handled by SIBYL.
- Chapter 5 covers the techiniques we have developed for retrieval of spoken documents and their use in QA.
- Chapter 6 describes the our named entity recogniser.
- Chapter 7 contains the methods an techniques used for Answer Extraction and ranking.
- Chapter 8 explains how coreference resolution is added to SIBYL.
- Finally, Chapter 9 concludes this work and gives perspectives about future research in this topic.
- The document is closed by Appendix A, containing the commented list of publications produced in this thesis.

2. Factoid Question Answering on Spoken Documents: State-of-the-Art

The interest on question answering on spoken documents is recent and there are few resources for research and evaluation on this topic. The state-of-the-art for speech question answering basically consists of systems evaluated in the QAst evaluation tracks (Question Answering on Speech Transcripts) [Turmo et al., 2007, 2008, 2009]. The QAst evaluation track at the CLEF workshop has provided a framework in which question answering systems can be evaluated in a real scenario, where the answers of both spontaneous oral questions and written questions have to be extracted from manually and automatically generated speech transcripts. This is the first and, until now, only framework of evaluation for speech question answering.

We start this Chapter giving a short historic perspective of the evolution of question answering, beginning with the first natural language interfaces to databases up to the processing of spoken documents (Section 2.1). In Section 2.2, the general architecture of a factoid question answering system is described and we review the QA literature. In Section 2.3 we describe the setting of the three QAst evaluations, detailing the corpus of spoken documents and questions of each year. Finally, Section 2.4 reviews the state-of-the-art for factoid question answering on speech transcripts that is derived from the QAst evaluations.

2.1 Pocket-sized History of QA on Spoken Documents

The history of Question Answering (QA from now on), probably begins with the question answering *routine* by Phillips [1960]. This system was designed to answer questions about a given text having the comprehension level of 6 years old children. The ROUTINE system starts by chunking the text sentences into noun-phrases and prepositional phrases. Then the chunks are stored as a list of five elements: subject, verb, object, time and place. The answering process consists in finding the best match for the analysed question from the set of analysed text sentences. This procedure is similar to searching information in a database with a natural language query.

The first published natural language interface for databases (NLIDB) is probably the BASEBALL system by B.F. Green in 1961 [Green et al., 1961]. This program was able to read a question written in ordinary English on punched cards and answer it. The domain was the 1958 American baseball league, restricted to the concepts scores, teams, month, day and place. BASEBALL could answer complex questions such as "*Where did each team play in July?*" but was strongly limited in the syntactic structures it could parse. The main advantage of this approach is that it hided the database structure from the user who does not need to learn a query language.

BASEBALL was the ancestor of a long line of NLIDBs of increasing complexity developed during the 70s and 80s. After these efforts, the field of QA did not evolve towards exploiting incrementally richer databases of relational information (with notable exceptions [Katz, 1997]), but in the line of ROUTINE to open-domain questions formulated on a collection of free text. The overview by Copestake and Spärck-Jones [1989] about NLIDBs pinpoints some of the reasons that prevented the widespread of commercial NLIDBs during the 80s. On the one hand, the sub-language those systems could accept was so restricted that it was not better than a traditional formal query language. On the other hand, any attempt to devise a sufficiently expressive sub-language was hindered by either ambiguity or limitations on coverage and usability. For these reasons and other factors (e.g., the development of graphic interfaces to access databases), the amount of published research on NLIDBs decreased abruptly during the 90s (as reported by Androutsopoulos et al. [1995]) until almost disappear, while the research on free text question answering increased. We even dare to point other reasons for this interest shift, from database-oriented QA to written text-oriented QA. Starting with personal computers massively marketed by IBM in the 80s and ending with the Internet breakthrough in the 90s, the evolution of computing encouraged an exponential explosion in the available amount of raw digital text. Nowadays, virtually all newspapers, books and personal documents are typeset with computers. The production of digital text has rapidly surpassed the amount of structured data stored in knowledge bases, while Internet has provided a widespread availability. Thus, it is clear that exploiting this digital text is worth the effort and its applicability to real-world problems seems guaranteed in the future.

During the last 15 years, the focus of research about QA has been mainly driven by the DARPA sponsored programs TIPSTER and AQUAINT, who have managed the Text REtrieval Conference (TREC) evaluations on QA and other related tasks. Many other useful advances have been achieved during this period, like better named entity recognisers and parsers or improvements in question analysis, which are very helpful for text processing.

In 2001, a committee of researchers from twenty institutions outlined a QA Roadmap for the TREC evaluation [Burger et al., 2001]. Having the ultimate goal of guiding the research towards a high-end QA system for real-world users, this document defines the types of real-world users and sets milestones and intermediate goals in the capabilities offered by the QA systems. It also describes the exact evolution of the QA evaluations in TREC during the 2001–2005 period. To our knowledge, this is the first time that multimedia QA is proposed in the literature. Burger et al. [2001] proposed to "*Extract answers from multimedia data*" as an eventual farthest step in the development of techniques for handling heterogeneous data sources. Although speech was not directly mentioned, they listed varied data formats such as databases, knowledge bases, and several document formats: SGML, PDF files, postscript files, Word, Excel and PowerPoint documents. Unfortunately, none of these varied data formats was really used in any of the following TREC evaluations.

Later in 2001, in the introduction to the Natural Language Engineering journal published by Cambridge Press, Hirschman and Gaizauskas state that "*We can even imagine applying question answering techniques to material in other modalities, such as annotated images or speech data*" [Hirschman and Gaizauskas, 2001]. This is the first explicit mention to question answering on speech data we are aware of.

Eight years later, open-domain QA systems where not yet as much developed as the road-maps where aiming to. In 2009, a group of USA researchers from the industry and academia gathered together to impulse an open collaboration in the development of integrated QA technologies. The OAQA (Open Advancement of Question Answering Systems) report [Ferrucci et al., 2009] is a new road-map that identifies general problems of the field, *inter alia*: 1) QA systems are too complex to replicate their results from the information presented in an academic paper, 2) there are no means to leverage individual contributions from distributed research groups, and 3) it is difficult to determine which technologies are really working.[1] Ferrucci et al. [2009] outlines collaboration strategies

[1] We subscribe these general problems of the field, too.

and guidelines for QA research, and also pinpoints in a footnote that "[...] *we limit our discussion to text-based QA, but acknowledge that QA systems can be developed to directly address other modalities including image, speech, music and video for example*."

In summary, many researchers have thought about QA on spoken documents, but it has always been left as future work in the main research initiatives.

Our work in spoken QA is rooted in the 2004-2006 period, with the European Commission funded CHIL project (Computers in the Human Interaction Loop, IST–2004–506909), which was aimed to develop context aware multimedia tools for providing a rich human-computer interaction in an intelligent meeting room.[2] Within the context of this project, we started working on a prototype capable of providing access to recorded audio, from the meetings previously held in the intelligent room, through a QA interface. Although this prototype system of QA on speech transcripts was not finally deployed, the CHIL project provided the starting funding to carry a public evaluation track in the 2007 CLEF evaluation campaign. The evaluation, named QAst, was aimed to research groups interested in the topic of question answering on spoken documents and lasted for three years (2007–2009). It was created together by three partners of the CHIL consortium: UPC (as a coordinator of the event), LIMSI[3] (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur) and ELDA (Evaluation and Language resources Distribution Agency). The following sections of this Chapter describe in detail these evaluations, covering the data sets, the methodological procedures and the approaches taken by the participants.

2.2 QA Methods and Architecture

In the TREC QA evaluation track, more than 200 systems have been evaluated over the years on the task of answering factoid and definitional questions. And at least 200 more have been evaluated in the QA@CLEF tracks, specially in cross-language scenarios. It is impossible to give proper comment of the full range of techniques and methods features in these systems in a reasonable space, but after reviewing this literature we can try to identify the bare bones that are common to most of their architectures.

2.2.1 Basic Architectures

The architecture of a factoid QA system can be reduced to a minimal schema that consists of three phases performed in a sequential pipeline [Turmo et al., 2009]. Figure 2.1 shows this architecture and the information exchange between the three modules: Question Processing, Passage Retrieval and Answer Extraction. This is a description of a generic QA architecture, specific systems may present differences.

[2] http://isl.ira.uka.de/fileadmin/templates/HTML/CHIL/servlet/is/101/index.html

[3] http://www.limsi.fr/Recherche/TLP/PageTLP.html

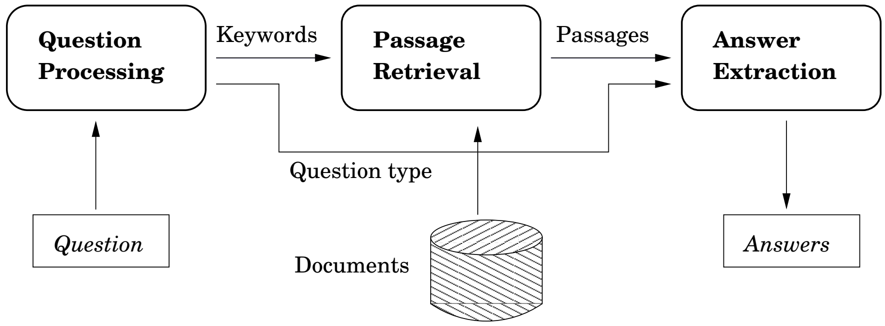


Figure 2.1: Overview of a generic QA architecture

1. **Question Processing**: Many factoid questions explicitly express a relation about the answer type they expect in the form of hyponym relation (i.e. *"What is the principal port in Ecuador?",* expects the name of a port, that's a kind of location; *"What gas is 78 percent of the earth's atmosphere?"* expects the name of a gas), and also several other relations describing its context (i.e. spatial, temporal, etc.). The Question Processing module analyses the question aiming to exploit some of these relations.

   It is frequent that the system guesses what is the expected answer type (EAT) in the form of a hyponymy relation. Most systems try to do this with hand-made lexical patterns [Prager et al., 2000], syntactic rules [Magnini et al., 2002], or machine learning classifiers [Punyakanok et al., 2004]. These expected answer types are mapped to either nodes from (parts of) an ontology [Hovy et al., 2001, Kwok et al., 2001] or to named entity types [Wu et al., 2005] (this is arguable a simplification of the ontology approach), to be later used in the Answer Extraction module.

   Additionally, almost any system uses some proceeding to select a subset of the question words (called *keywords*) to be used later in the Passage Retrieval phase.

2. **Passage Retrieval**: The second module is an Information Retrieval module. It retrieves documents or shorter passages from the documents collection using the keywords extracted by the Question Processing module. Ideally, these passages should contain the answer to the question. The documents collection may be either a set of documents fixed for the evaluation (like the AQUAINT collection [Graff, 2002] in the TREC evaluations) or the Internet. These collections are accessed by posting queries to local IR engines or to commercial web search engines.

   Many systems find alternations to these keywords (query expansion), in the form of synonyms or semantically related words to increase the coverage of the retrieval phase. For example, this can be achieved by adding new keywords to the original query after performing a blind relevance feedback loop on the Internet [Yang and Chua, 2002]. It is also frequent to filter out the results not having a lexical or semantic relatedness to the query in order to increase precision [Vicedo et al., 2002]. Additionally, an accurate passage retrieval helps to reduce the computational overhead of the Answer Extraction module.

3. **Answer Extraction**: Finally, the answer extraction module extracts exact answers from the retrieved passages. Usually this involves two different processes: first a set of *candidate answers* are identified and then the answer is selected from the candidates.

   Generating a set of candidate answers is usually dependant on the selected EAT typology. For example, if the system classifies the EATs as named entities types, a NERC is enough to generate the candidate answers list, e.g. any person name would be a candidate answer to question *"Who is the world's best football player?."* Usually, the considered candidate answers are either named entities, or noun and verb phrases [Sun et al., 2005], or just any sequence of words [Brill et al., 2002, Clarke et al., 2002], that fulfils some given conditions.

   The answer selection may be a complex mechanism involving several layers of filtering and scoring functions. In the general case, it outputs an ordered list of the n most probable answers to the input questions given the information extracted from the question in the first phase.

It is also usual to apply some kind of document preprocessing prior to question answering. This off-line annotation dramatically reduces system's computation time. For factoid QA the most usual is to use a named entity recogniser and classifier to identify all named entities occurring in the text. Additional information, such as part-of-speech tagging or syntactic parsing may be necessary information for further steps in the QA process.

2.2.2 Answer Extraction Mechanisms

The Answer Extraction part is the most elaborated and variable part of the simple QA pipeline we have presented. The two most frequently approaches we can find in the literature are based on statistical methods or on linguistic methods.

The statistical methods exploit the massive amounts of data available on the Internet to perform a redundancy-based QA. Instead of using sophisticated natural language processing, they rely on information quantity to overcome the lack of quality. The core mechanism is to estimate what phrases are more commonly occurring together with the question terms [Dumais et al., 2002, Lin and Katz, 2003].

The linguistic methods work in a more fine-grained manner than statistical methods. Usually they operate with a single sentence where the candidate answer realises and try to determine whenever this sentence is stating what it is requested in the question. We classify the most frequently used linguistic methods into two groups depending on the their approach to the task. They can be based either in calculating a value for the *similarity* between question and answer sentence, or in calculating a value for the *answerness* of the sentence containing the candidate answer. And in most cases these measures are calculated from the perspective of either *lexical* information (or surface realisation), or *syntactic* information. The combination of both factors yields four groups of methods that try to assess one of the two measures from one of the two perspectives:

- The idea behind *similarity* is that the question is mentioning some entities, referring to some events, and expressing some kind of restrictions that define a (potentially) single unique entity that is the answer. The more similar are the question and the sentence structures, the more likely are to convey the same meaning. Thus the extraction process consists in evaluating the similarity of the context surrounding the candidate answer with information extracted from the question.

  The similarity is more frequently evaluated under a syntactic perspective. Syntax allows to check relations between phrases and answers (like the presence of a temporal modifier) disregarding the exact lexical realisation of the phrase. Usually the system checks if the whole syntax of the question is similar to the syntax of the sentence [Sun et al., 2005, Moschitti and Quarteroni, 2010], but it is possible to do this for only certain selected keywords [Aktolga et al., 2011].

  Evaluating the surface similarity between question and answer context is possible with simple string matching techniques or with more elaborated strategies. For example, turning the question into a declarative statement and then try to locate this statement (or paraphrases of it) in the collection [Hermjakob et al., 2002]. Simple measures of keyword occurrences, word distance and appositions may help to also capture frequent constructions found in the answers [Moldovan et al., 1999].

- In contrast, *answerness* consists in determining to what extend a sentence is stating a fact in the typical form of doing so for this type of fact. Taking advantage of the corpus redundancy (specially in Internet), the system can attempt to obtain the answers only if their are expressed in certain previously learnt forms. These forms can be lexical *surface forms* or *syntactic forms*. These *answerness* patterns are typically associated to the EAT typology that the QA system uses.

  These surface realisations can be word patterns like in this example: `discovery of NAME by ANSWER`. This concrete pattern allows to detect who has discovered something bearing a name that has been extracted from the question. These surface realisations are not so closely related to the question structure as in the surface similarity described in the previous item, for example, consider the equivalent pattern `ANSWER was invented by NAME in DATE`. The surface realisations may have been obtained by creating hand-made patterns [Chali and Dubien, 2007] or can be automatically acquired from diverse sources: mining the web for realisation examples that are then turned into regular expressions [Ravichandran and Hovy, 2002]; automatically adapting realisations extracted from FrameNet [Kaisser et al., 2006]; using a statistical machine translation system trained for the question/answer language pair in order to recognise [Echihabi and Marcu, 2003] or to generate [España-Bonet and Comas, 2012] the answer patterns.

  It is also possible to learn what syntactic forms are more frequently associated to the realisation of some EAT types [Kaisser, 2012]. For example (from [Kaisser, 2012]), the questions of the form `When+was+NP+VERB?` are frequently answered by sentences containing these two dependency relation paths: `1:NP↑pobj↑prep↑nsubj↓prep↓pobj`, and `2:VERB↑nsubj↓prep↓pobj`.

2.2.3 Beyond the Basic Architecture

Almost all QA systems differ at some point from the basic architecture we have shown in Figure 2.1. Specially they differ in terms of what information is used in each module and how it flows, besides, they may include several kinds of feedback loops and fall-back mechanisms between modules (e.g., re-run Passage Retrieval with different parameters if the Answer Extractor is unable to find any candidate answers). Beyond this basic architecture, more complex systems can be found in the literature. Here we report a few of the most frequent additions:

- Some QA systems have added other modules after the Answer Extraction aiming to validate the output. For example, checking it against other information sources, or using the notion of entailment [Wang and Neumann, 2007b, Harabagiu and Hickl, 2006] between question and answer document, or introducing a certain amount of reasoning produced by simple syntax-rewriting rules [Bouma et al., 2005].
- It is also possible to have several different full-fledged QA systems arranged in parallel under a common architecture and combine their outputs into a single stream of answers. The rationale of this process is that systems that employ different answering techniques will have different strengths and weaknesses according to question types. Combining the output of multiple QA systems can be achieved with simple strategies, *inter alia*: voting or ordering by likelihood [Prager et al., 2003], automatically learning how to boost the weight of the most confident system for each particular question type [Jijkoun and De Rijke, 2004, Ko et al., 2007], or selecting the best answers considering the relations between them as new evidences [Dalmas and Webber, 2007, Mendes and Coheur, 2011].
- Another possibility is to expand the architecture by adding a layer of question analysis before the QA system in order to decompose the questions into simpler sub-questions that handle specific topics (i.e. time and space restrictions) and combine the answers of the sub-questions to rewrite the original question into an easier one [Saquete et al., 2004, Hartrumpf et al., 2009, Kalyanpur et al., 2011b].
- Integrating QA with Machine Translation leads to cross-lingual QA systems that can find answers in documents written in languages different from the question and translate them, thus yielding more complex architectures [Bos and Nissim, 2006, Sacaleanu et al., 2007].
- Integrating QA with dialog systems can lead to interactive systems with an interface suitable for providin disambiguation, answer justification, and error explanation to the user [Sonntag, 2009, Dornescu, 2010, Dang et al., 2007].
- There are systems that use more advanced extraction mechanisms than the ones we have commented in the basic architecture; knowledge intensive QA systems that use in-deep linguistic and logic tools. It is possible to collate world-knowledge sources [Moldovan and Rus, 2001] and derive proposition logic from the text representation

in order to use theorem provers to demonstrate the logical implication between question and answer [Moldovan et al., 2007b, Furbach et al., 2010, Babych et al., 2011]. Knowledge intensive systems usually get better results than the more basic approaches.

Outside the research environments provided by the CLEF, TREC, and other evaluation forums, the most noticeable impact any QA application has ever had in the media is the success achieved by WATSON, the open-domain QA system developed by IBM [Ferrucci et al., 2010]. In February 2011, WATSON competed in an american television quiz show called *Jeopardy*, and defeated the best human players at the game of answering open-domain riddle-like questions. WATSON's technology consists of a super-computer running massively parallel ensembles of QA technologies, and its architecture resembles the one of the last IBM's TREC system, PIQUANT II [Chu-Carroll et al., 2004]. WATSON uses a precompiled knowledge source to answer the questions. It is remarkable that at least an 81.3% of the Jeopardy questions' answers (from an historical archive) are the title of a Wikipedia page [Chu-Carroll and Fan, 2011]. Since Jeopardy questions are of an encyclopedic nature, WATSON's documents collection contains not only the full English Wikipedia but several other hand-picked encyclopedias, dictionaries and structured data sources. Additionally, WATSON uses a *source expansion* algorithm that enriches the encyclopedia entries with new facts automatically collected from the Internet. Thus, obtaining a collection of documents whose titles are the answers of, at least, 89.17% of the questions [Schlaefer et al., 2011]. To answer a questions, WATSON retrieves a list of relevant entities from the encyclopedias and filters them using their associated facts and several knowledge-based and statistically-based methods. Many of the filters are not based in knowledge extracted from the documents collection, but rather are based in using multiple ontologies in parallel [Kalyanpur et al., 2011a], or learning patterns from massive amounts of text [Fan et al., 2011]. The methods and algorithms implemented in WATSON for generating and filtering the candidates have no evident application to the problem of QA on spoken documents, since it works with precompiled encyclopedic knowledge or knowledge collated from structured sources obtained from outside the spoken documents.

In this dissertation we report about the design of SIBYL, our factoid QA system for spoken documents. Following the classification we have applied in this section, we can put it in relation to the rest of the state-of-the-art:

- SIBYL uses a named entity-based EAT typology and a NERC to identify candidate answers in the transcripts.

- For the Passage Retrieval we have deployed a specially designed IR engine tailored for transcripts.

- The Answer Extraction mechanism is based in two measures: the former is a simple surface measure based on keyword density around the candidate, while the latter is a learnable estimation of syntactic similarity between question and answer.

- Beyond the basic architecture, Sibyl makes use of a coreference resolution module that is used to improve the coverage of Passage Retrieval and Answer Extraction.
- Sibyl does not use information sources other than the documents collection, since the aim of this work is to investigate speech phenomenons rather than improving general open-domain QA.

2.3 The QAst Evaluation Framework

One of the objectives of this thesis has been to help and promote the creation of evaluation frameworks for the task of question answering on transcripts of spoken documents. To our knowledge, there were no resources related to this problem prior to year 2007 and no published works explicitly dealing with it. This section provides a detailed description of the QAst data-sets and results achieved in the evaluation.

The QAst evaluation frameworks should provide public available benchmarks of test data for question answering on speech transcripts. These benchmarks fulfil a two-fold objective. On the one hand, to make comparable the methods and results obtained across the researchers, in a similar way to the TREC question answering tasks.[4] On the other hand, they to help to define and settle this task and to develop an state-of-the-art of this topic.

A suitable evaluation framework for spoken QA should allow to study the relevance of two important factors in question answering: the effect of loose syntax in speech, and the effect of speech recognition errors. Thus, the ideal document collection would be a large collection of human-made transcripts with corresponding automatic transcriptions obtained using Automatic Speech Recognition (ASR) having, if possible, several ASRs of different word error rates (WER).

As a result of this effort the QAst task was created, which stands for Question Answering on Speech Transcripts,[5] as a pilot task in the CLEF conference. This task lasted for three years, from 2007 to 2009 [Turmo et al., 2007, 2008, 2009]. Each year a different set of question answering benchmarks was released, comprising several languages and scenarios of usage. The answers were manually assessed by the Evaluation and Language resources Distribution Agency (ELDA)[6] and finally published with the questions.

We have participated in the three QAst evaluations with the upc team. In these evaluations we have used several baseline versions of Sibyl [Comas et al., 2007, Comas and Turmo, 2008b, 2009] that later have been further improved. The baseline systems follow the basic three-module architecture described in Section 2.2 and use named entities as candidate answers.

[4] http://trec.nist.gov/data/qa.html
[5] http://www.lsi.upc.edu/~qast
[6] http://www.elda.fr

The same evaluation measures were used in all three QAst evaluations, we describe these measures in the next section. In Sections 2.3.2 to 2.3.4 we describe the data sets and results of each QAst evaluation. We do not comment the results of the French evaluations from 2008 and 2009 editions and refer the reader to the official overviews for this information [Turmo et al., 2007, 2008, 2009].

2.3.1 Evaluation Measures

In all QAst evaluations the QA system are entitled to output a ranking of at most 5 answers for each question. An answer in the output ranking is considered correct by the human evaluators if it contains the complete and exact answer, and it is supported by the corresponding document. If an answer is incomplete or it includes more information than necessary or the document does not provide the justification for the answer, the answer is considered incorrect. Correct answers are evaluated with two measures:

- *Mean Reciprocal Rank (MRR):* Average of inverses of the ranking of the first correct answer for each question. It is defined as $\frac{1}{N}\sum_{i=0}^{N}\frac{1}{rank_i}$, where $rank_i$ is the position of the first correct answer in the answer's list for question number i. When the considered length of the answer list is n, it is also referred as MRR@n.
- *Accuracy:* The fraction of correct answers ranked in the first position in the list of 5 possible answers.

Questions without answer in the text are evaluated in the same way as the other questions: "nil" is the correct answer and it may be anywhere in the ranking. Then MRR and accuracy are calculated as defined.

QAst evaluations also report Top1 and Top5 measures. Top1 is the number of questions that have a correct answer ranked first, and Top5 denotes how many have a correct answer anywhere in the 5 answers ranking. In 2008 and 2009 evaluation, these measures are calculated separately for factoid and definitional questions.

2.3.2 QAst 2007 Evaluation

In the first QAst edition from CLEF 2007 [Turmo et al., 2007], four English monolingual tasks were defined as follows:

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

2.3.2.1 Documents and Questions

The data sets for this evaluation consist of two different resources, one for the lecture scenario (CHIL corpus) and one for the meeting scenario (AMI corpus):

- The CHIL corpus:[7] it consists of 25 one hour lectures both manually and automatically transcribed. The domain of the lectures is *speech and language processing*. LIMSI produced the ASR transcriptions with around 20% of WER [Lamel et al., 2005]. In addition, the set of lattices and confidences for each lecture were provided. The language is European English (mostly spoken by non-native speakers).
- The AMI corpus:[8] it consists of around 100 hours (168 meetings) both manually and automatically transcribed. The domain of these meetings is *design of television remote control*. The University of Edinburgh produced the ASR transcripts with around 38% of WER [Hain et al., 2007]. Four people take part in the meeting. The language is European English.

All the questions in this QAst evaluation are factoid questions, whose expected answers are named entities ('person', 'location', 'organisation', 'language', 'system', 'method', 'measure', 'time', 'colour', 'shape' and 'material'). The two data collections (CHIL and AMI corpora) 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. Correct answers are provided for all questions. An answer is composed of an answer-string (that contains nothing more than a complete and exact answer, i.e. a named entity) and the unique identifier of a document that supports the answer. Two sets of questions were provided for each scenario, one for development and one for test. The document collections are also split into development and test parts:

- Development set:
  - Lectures: 10 seminars and 50 questions.
  - Meetings: 50 meetings and 50 questions.
- Evaluation set:
  - Lectures: 15 seminars and 100 questions.
  - Meetings: 118 meetings and 100 questions.

2.3.2.2 Evaluation Results

Five different teams participated in this evaluation, namely:

[7] http://chil.serve.de

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

Transcript	System	Questions	Top5	MRR	Accuracy
Manual	CLT1	98	16	0.09	0.06
	CLT2	98	16	0.09	0.05
	DFKI1	98	19	0.17	0.15
	LIMSI1	98	43	0.37	0.32
	LIMSI2	98	56	0.46	0.39
	TOKYO1	98	32	0.19	0.14
	TOKYO2	98	34	0.20	0.14
	UPC1	98	54	0.53	0.51
ASR	CLT1	98	13	0.06	0.03
	CLT2	98	12	0.05	0.02
	DFKI1	98	9	0.09	0.09
	LIMSI1	98	28	0.23	0.20
	LIMSI2	98	28	0.24	0.21
	TOKYO1	98	17	0.12	0.08
	TOKYO2	98	18	0.12	0.08
	UPC1	96	37	0.37	0.36
	UPC2	97	29	0.25	0.24

Table 2.1: Results of 2007 tasks T1 and T2: transcriptions of CHIL seminars

Transcript	System	Questions	Top5	MRR	Accuracy
Manual	CLT1	96	31	0.23	0.16
	CLT2	96	29	0.25	0.20
	LIMSI1	96	31	0.28	0.25
	LIMSI2	96	40	0.31	0.25
	UPC1	95	27	0.26	0.25
ASR	CLT1	93	17	0.10	0.06
	CLT2	93	19	0.13	0.08
	LIMSI1	93	21	0.19	0.18
	LIMSI2	93	21	0.19	0.17
	UPC1	91	22	0.22	0.21
	UPC2	92	17	0.15	0.13

Table 2.2: Results of the 2007 task T3 and T4: transcriptions of AMI meetings

- CLT: Center for Language Technology, MacQuairie University, [Mollá et al., 2007].
- DFKI: German Research Centre for Artificial Intelligence [Neumann and Wang, 2007].
- LIMSI: Spoken Language Processing Group, LIMSI-CNRS, [Rosset et al., 2007].
- TOKYO: Tokio Institute of Technology, [Whittaker et al., 2007].
- UPC: Technical University of Catalonia, UPC, [Comas et al., 2007].

Tables 2.1 and 2.2 show the results of this evaluation. The results obtained with lectures were much better than with meetings, showing that the two corpora were different in nature. We participated with two runs, UPC1 and UPC2 [Comas et al., 2007]. One was the baseline version of SIBYL, with none of the extensions for named entity recognition, passage retrieval or answer extraction that are described in this dissertation, while the other was the baseline system with our PHAST IR engine (Chapter 5) for automatic transcripts. Definitional questions were handled as factoid questions by SIBYL, since we had no specific modules for them. SIBYL obtained the best overall results with lecture transcripts and automatic transcripts of meetings.

2.3.3 QAst 2008 Evaluation

A total of five tasks are defined for the second edition of QAst [Turmo et al., 2008], covering five main task scenarios and three languages: English, Spanish and French. T1 and T2 tasks use the same document collections from QAst 2007:

- T1: QA in transcriptions of lectures in English.
- T2: QA in transcriptions of meetings in English.
- T3: QA in transcriptions of broadcast news for French.
- T4: QA in transcriptions of European Parliament Plenary sessions in English.
- T5: QA in transcriptions of European Parliament Plenary sessions in Spanish.

2.3.3.1 Data and Questions

The 2008 data is derived from five different resources, covering spontaneous speech, semi-spontaneous speech and prepared speech:

- The CHIL corpus (as used for QAst 2007).
- The AMI corpus (as used for QAst 2007).

- French broadcast news: The test portion of the ESTER corpus [Galliano et al., 2006] contains 10 hours of broadcast news in French, recorded from different sources. There are 3 different automatic speech recognition outputs with different WER (i.e., 11.0%, 23.9% and 35.4%).
- English parliament: The TC-STAR05 EPPS English corpus,[9] which contains 3 hours of recordings from the European Parliament in English (about 35,000 words long). The data was used to evaluate speech recognisers in the TC-STAR project (IST–2002–FP6-506738). There are 3 different automatic speech recognition outputs with different word error rates (10.6%, 14.0% and 24.1%, respectively).
- Spanish parliament: The TC-STAR05 EPPS Spanish corpus[9] is comprised of three hours of recordings from the European Parliament in Spanish. The data was used to evaluate recognition systems developed in the TC-STAR project. There are 3 different automatic speech recognition outputs with different word error rates (11.5%, 12.7% and 13.7%), obtained from the TC-STAR evaluation.

The spoken data covers a broader range of types, both in terms of content and in speaking style, than the QAst 2007 data. The Broadcast News data is less spontaneous than the lecture and meeting speech as they are almost read and are closer in structure to written texts. The EPPS data lies between prepared speech and spontaneous speech. It is prepared because the speakers had previously planned what they would say, but they do not have a full written reference text. There is a significant amount of improvisation in the speech. While meetings and lectures are representative of spontaneous speech, Broadcast News and European Parliament sessions are usually referred to as prepared speech. Although they typically have few interruptions and turn-taking problems if compared to meeting data, many of the characteristics of spoken language are still present (hesitations, breath noises, speech errors, false starts, mispronunciations and corrections).

Two types of questions are considered in QAst 2008: factoid questions and definitional ones. For each corpus (CHIL, AMI, ESTER, EPPS EN, EPPS ES) roughly 70% of the questions are factoid, 20% are definitional, and 10% are "nil" (i.e., questions having no answer in the document collection). The factoid questions are similar to those used in the 2007 evaluation, the expected answer to these questions is a named entity ('person,' 'location', 'organisation', 'language', 'system', 'method', 'measure', 'time', 'colour', 'shape,' and 'material'). The definition questions are questions such as *What is the Vlaams Blok?* and the answer must be some kind of definition provided in the documents such as *a criminal organization* or *political groups* . The definition questions focus on defining one of these entity types: person, organisation, object, other.

The provided answers for both factoid and definitional questions consist of an answer-string (that contains nothing more than a complete and exact answer, i.e. a named entity) and the unique identifier of a document that supports the answer.

For each of the five scenarios, two sets of questions have been provided to the participants, the first for development purposes and the second for evaluation. This is the data-sets breakdown:

[9] `http://www.tc-star.org`

		Factoid			Definitional			All	
Transcript	**System**	**Top5**	**MRR**	**Acc.**	**Top5**	**MRR**	**Acc.**	**MRR**	**Acc.**
Manual	LIMSI1	29	0.32	29.3	13	0.44	36.0	0.35	31.0
	LIMSI2	29	0.32	29.3	13	0.42	32.0	0.35	30.0
	UPC1	9	0.11	9.3	3	0.05	0.0	0.09	7.0
	LIMSI1	20	0.25	24.0	8	0.28	24.0	0.26	24.0
ASR A	UPC1	5	0.05	4.0	0	0.00	0.0	0.04	3.0
11.5% WER	UPC2	5	0.06	5.3	2	0.08	8.0	0.07	6.0
	LIMSI1	18	0.20	17.3	9	0.28	24.0	0.22	19.0
ASR B	UPC1	5	0.06	5.3	0	0.00	0.0	0.05	4.0
12.7% WER	UPC2	5	0.06	5.3	2	0.08	8.0	0.07	6.0
	LIMSI1	20	0.24	22.7	8	0.27	24.0	0.25	23.0
ASR C	UPC1	2	0.03	2.7	0	0.00	0.0	0.02	2.0
13.7% WER	UPC2	3	0.03	2.7	1	0.04	4.0	0.04	3.0

Table 2.3: Results of the 2008 task T5: Spanish EPPS transcripts (75 factoid questions and 25 definitional ones)

		Factoid			Definitional			All	
Transcript	**System**	**Top5**	**MRR**	**Acc.**	**Top5**	**MRR**	**Acc.**	**MRR**	**Acc.**
Manual	CUT1	14	0.18	17.9	2	0.09	9.1	0.16	16.0
	CUT2	16	0.19	16.7	8	0.26	18.2	0.20	17.0
	LIMSI1	48	0.53	47.4	4	0.18	18.2	0.45	41.0
	UPC1	39	0.44	38.5	4	0.18	18.2	0.38	34.0
ASR	LIMSI1	33	0.34	30.8	3	0.14	13.6	0.30	27.0
20% WER	UPC1	35	0.39	34.6	4	0.18	18.2	0.34	31.0
	UPC2	35	0.37	33.3	4	0.18	18.2	0.33	30.0

Table 2.4: Results of the 2008 task T1: English lectures (78 factoid questions and 22 definitional questions)

		Factoid			Definitional			All	
Transcript	**System**	**Top5**	**MRR**	**Acc.**	**Top5**	**MRR**	**Acc.**	**MRR**	**Acc.**
Manual	LIMSI1	44	0.47	37.8	7	0.22	19.2	0.40	33.0
	UPC1	29	0.35	31.1	3	0.12	11.5	0.29	26.0
ASR	LIMSI1	23	0.21	16.2	6	0.18	15.4	0.20	16.0
38% WER	UPC1	19	0.20	17.6	5	0.19	19.2	0.20	18.0
	UPC2	16	0.16	10.8	6	0.23	23.1	0.18	14.0

Table 2.5: Results of the 2008 task T2: English meetings (74 factoid questions and 26 definitional questions)

- Development set:
  - Lectures: 10 seminars and 50 questions.
  - Meetings: 50 meetings and 50 questions.
  - French broadcast news: 6 shows and 50 questions.
  - English EPPS: 2 sessions and 50 questions.
  - Spanish EPPS: 2 sessions and 50 questions.
- Evaluation set:
  - Lectures: 15 seminars and 100 questions.
  - Meetings: 120 meetings and 100 questions.
  - French broadcast news: 12 shows and 100 questions.
  - English EPPS: 4 sessions and 100 questions.
  - Spanish EPPS: 4 sessions and 100 questions.

2.3.3.2 Evaluation Results

Five different teams participated in the 2008 evaluation, namely:

- cut: Dept. Computer Science and Media, Chemnitz University, [Kürsten et al., 2008].
- inaoe: Laboratorio de Tecnologías del Lenguaje, Instituto Nacional de Astrofísica, Optica y Electrónica (INAOE).
- limsi: Spoken Language Processing Group, LIMSI-CNRS, [Rosset et al., 2008].
- ua: Dept. of NLP and Information Systems, University of Alicante [Pardiño et al., 2008].
- upc: Technical University of Catalonia, [Comas and Turmo, 2008b].

In this evaluation (Table 2.3), we focused on porting Sibyl to the Spanish language for the new EPPS track. But our effort was hindered by very poor named entity recognition and the relatively low performance of our Spanish question classifier, that dropped to only 74% in the QAst questions. Out of the 75 factoid questions, our NERC was able to label the answer with the correct named entity type in only 8 of the retrieved answers, thus yielding a very poor answer recall.

Our system for the English language scenarios (Tables 2.4, 2.5, 2.6) was very similar to the previous year system (QAst 2007), with only minor tweaking and bug correction. Like in the previous evaluation, nothing special was done to handle definitional questions in Sibyl.

		Factoid			Definitional			All	
Transcript	**System**	**Top5**	**MRR**	**Acc.**	**Top5**	**MRR**	**Acc.**	**MRR**	**Acc.**
Manual	CUT1	12	0.16	16.0	9	0.36	36.0	0.21	21.0
	CUT2	12	0.16	16.0	11	0.39	36.0	0.22	21.0
	INAOE1	41	0.43	37.3	6	0.21	20.0	0.38	33.0
	LIMSI1	44	0.43	33.3	12	0.39	32.0	0.42	33.0
	UA1	32	0.30	21.3	4	0.16	16.0	0.27	20.0
	UPC1	38	0.44	40.0	4	0.16	16.0	0.37	34.0
ASR A 10.6% WER	INAOE1	32	0.37	33.3	5	0.20	20.0	0.33	30.0
	INAOE2	34	0.38	32.0	5	0.20	20.0	0.33	29.0
	LIMSI1	24	0.23	18.7	9	0.31	28.0	0.25	21.0
	UA1	12	0.09	4.0	4	0.16	16.0	0.10	7.0
	UPC1	18	0.22	20.0	4	0.17	16.7	0.21	19.0
	UPC2	16	0.16	13.3	4	0.17	16.7	0.16	14.1
ASR B 14.0% WER	LIMSI1	22	0.21	16.0	9	0.33	32.0	0.24	20.0
	UA1	12	0.11	8.0	4	0.16	16.0	0.12	10.0
	UPC1	15	0.18	16.0	4	0.16	16.0	0.17	16.0
	UPC2	14	0.16	13.3	4	0.16	16.0	0.16	14.0
ASR C 24.1% WER	LIMSI1	21	0.21	16.0	8	0.30	28.0	0.23	19.0
	UA1	9	0.10	8.0	5	0.20	20.0	0.12	11.0
	UPC1	11	0.11	9.3	5	0.20	20.0	0.14	12.0
	UPC2	11	0.11	8.0	4	0.16	16.0	0.12	10.0

Table 2.6: Results of the 2008 task T4: English EPPS transcripts (75 factoid questions and 25 definitional ones)

2.3.4 QAst 2009 Evaluation

A total of six tasks are defined for this QAst [Turmo et al., 2009], covering three languages and two kinds of questions:

- T1a: QA of English written questions in the manual and automatic transcriptions of European Parliament Plenary sessions in English (EPPS English corpus).
- T1b: QA of manual transcriptions of English spontaneous oral questions in the manual and automatic transcripts of the EPPS English corpus.
- T2a: Same as T1a task but using the EPPS Spanish corpus.
- T2b: Same as T1b task but using the EPPS Spanish corpus.
- T3a: Same as T1a task but using transcriptions of French broadcast news (the ESTER corpus).
- T3b: Same as T1b task but using the ESTER corpus.

Type	Factoid	Definition	"nil"
T1 (English)	75%	25%	18%
T2 (Spanish)	55%	45%	23%
T3 (French)	68%	32%	21%

Table 2.7: Distribution of question types per task: T1 (EPPS EN), T2 (EPPS ES), T3 (ESTER).

2.3.4.1 Spoken Document Collection

The aforementioned three collections (T1, T2 and T3) are the same than the ones used for the QAst 2008 evaluation campaign: EPPS EN, EPPS ES and ESTER corpora. Additionally, each word in the automatic transcripts has an associated time-stamp, referring to the time-span in the audio recording when they are uttered (the audio files where not supplied for any of the QAst evaluations).

As for the previous year, two types of questions were considered: factoid and definitional. The expected answer to a factoid question is a named entity of type: 'person,' 'organisation,' 'location,' 'time,' and 'measure.' This is less than the 10 categories used for the 2007 and 2008 evaluations. Some categories were not considered in 2009 because no occurrences were found in the collected set of spontaneous questions ('language', 'system', 'colour', 'shape' and 'material'). Each task has a set of 50 development questions and 100 test questions.

For the manual transcripts, the answers provided consist of a pair of answer-string and the unique identifier of a document that supports the answer. The answer-string contains a complete and exact answer, i.e. a named entity). For the automatic transcripts, the answer consists of a time interval formed by a pair time-stamps and document id. In contrast with the 2008 task, these answers are valid for all three automatic transcripts at the same time, since they are not related to the actual transcript words but to the moment where the answer was uttered in the audio record.

For each language, a number of "nil" questions (i.e., having no answer in the document collection) were selected. The distribution of the different types of questions across the three collections is shown in Table 2.7.

We have used the English language task T1 as a test bed for most of the evaluations detailed in this dissertation, since this is the most complete English task and it was attempted by all QAst 2009 participants. Thus, in the rest of this dissertation we will refer to the individual sub-tasks of T1 as:

- Task *M*, which uses the collection of manual transcripts of recorded European Parliament Plenary Sessions (EPPS EN).
- Tasks *asrA*, *asrB*, and *asrC*, which correspond to the three different automatic transcripts of EPPS EN. These transcripts have an increasing level of word error rate: 10.6%, 14.0%, and 24.1%.

M: *Abidjan is going going the way of Kinshasa Kinshasa which was of course a country in the past with skyscrapers and boulevards and now a country a city in ruins*

asrA: *average down is going to go in the way of Kinshasa other at Kinshasa which was of course a country in the past of skyscrapers and poorer parts and our country as a city in ruins*

asrB: *other German is going to go in the way of Kinshasa happy at Kinshasa which was of course country in the past and skyscrapers and boulevards and in our country as a city in ruins*

asrC: *average down is going to going the way of kinshasa and acting shasta which was of course a country in the past the skyscrapers and boulevards and our country as a city in ruins*

Figure 2.2: Sample of manual (*M*) and automatic transcripts

Figure 2.2 shows a text sample extracted from the EPPS EN collection. As it can be seen, the distribution of recognition errors in each ASR is completely different from the others. Thus, the word error rate measure has not to be taken as a strictly *incremental* addition of noise over the previous transcript, but as a *different distribution and amount* of noise for each different ASR. He believe the distribution of errors is more important than the number of errors in QA.

2.3.4.2 Spontaneous Oral Questions

In QAst 2008 the questions were generated by human assessors who read through the documents and wrote questions answerable with the information found in the text. A novel feature in QAst 2009 was the introduction of spontaneous oral questions. The main issue in the generation of this kind of questions was how to obtain spontaneity [Buscaldi et al., 2009]. The solution adopted was to set up the following procedure for question generation:

1. Passage generation: a set of passages was randomly extracted from the document collection. A single passage was composed by the complete sentences included in a text window of 720 characters.

2. Question generation: human question generators were randomly assigned a number of passages (varying from 2 to 4). They had to read each passage and then formulate one or more questions about any of the passage topics, provided that the answer was not stated in the passage.

3. Question transcription: precise manual transcriptions of the oral spontaneous questions were made, including hesitations, etc. For instance, "*(%hesitation) What (%hesitation) house is the pres() the president elect being elected to?*"

4. Question filtering: some questions were filtered out from the set of generated questions because their answer types were not allowed or because they did not have answer in the document collection. The resulting questions were *usable* questions.

5. Written question generation: the usable questions were re-written by removing speech disfluencies, correcting the syntax and simplifying the sentence when necessary. For instance, "*What house does the president run?*"

6. Question selection: the final set of development questions and test questions were selected from the usable questions.

As a results, two question sets are formed. Set B contains oral spontaneous questions manually transcribed, while set A consists of grammatically corrected transcripts of the questions in set B (Buscaldi et al. [2009] provide more details about the process). All the experiments detailed in this dissertation refer to question set A.

As it has been pointed out in [Bernard et al., 2010], this method for gathering spontaneous questions used in QAst 2009 yields questions its answer is further in the text from the context words than in QAst 2008 questions. Thus, spontaneous questions are more difficult to answer for QA systems relying on shallow statistical measures of distance or relevance. This partially explains why the results of QAst 2009 where worse than QAst 2008 for the tasks using the same document collections.

2.3.4.3 Evaluation Results

In this edition, a total of four systems were evaluated:

- INAOE: Laboratorio de Tecnologías del Lenguaje, Instituto Nacional de Astrofísica, Optica y Electrónica (INAOE), [Reyes-Barragán et al., 2009].

- LIMSI: Spoken Language Processing Group, LIMSI-CNRS, [Bernard et al., 2009].

- TOKIO: Tokyo Institute of Technology, [Heie et al., 2009].

- UPC: Technical Univertisy of Catalonia, [Comas and Turmo, 2009].

The results of the English EPPS task are summarised in Tables 2.8 (manual transcripts) and 2.9 (automatic transcripts) for both question sets.[10] It shows accuracy, MRR, Top1 and Top5 scores for each track and run as defined previously. It must be noted that INAOE's numbers are difficult to compare with the other results: as stated in [Reyes-Barragán et al., 2009], INAOE enriched the *asrC* transcripts with named entities extracted from the *asrA*

[10] Note that in the official QAst results from [Turmo et al., 2009], the English EPPS task for written questions is named "T1a."

		Factoid			Definitional			All	
System	Q.Set	Top5	MRR	Acc.	Top5	MRR	Acc.	MRR	Acc.
INAOE1	A	44	0.38	26.7%	10	0.31	28.0%	**0.36**	27%
	B	28	0.27	21.3%	7	0.26	24.0%	0.27	22%
INAOE2	A	42	0.38	28.0%	9	0.30	28.0%	**0.36**	**28%**
	B	38	0.35	25.3%	9	0.30	28.0%	0.34	26%
LIMSI1	A	42	0.39	29.3%	11	0.28	20.0%	**0.36**	27%
	B	39	0.36	25.3%	10	0.24	16.0%	0.33	23%
LIMSI2	A	32	0.31	22.7%	13	0.36	24.0%	0.32	23%
	B	30	0.26	18.7%	11	0.30	20.0%	0.27	19%
TOK1	A	11	0.10	6.7%	3	0.03	0.0%	0.08	5%
	B	11	0.08	4.0%	3	0.03	0.0%	0.06	3%
UPC1	A	32	0.27	18.7%	8	0.29	28.0%	0.28	21%
	B	27	0.23	14.8%	9	0.320	30.7%	0.26	19%
UPC2	A	35	0.31	22.7%	8	0.29	28.0%	0.31	24%
	B	26	0.24	17.3%	9	0.320	32.0%	0.26	21%

Table 2.8: Results of the 2009 task T1: English EPPS transcripts (75 factoid questions and 25 definitional ones)

and *asrB* transcripts because of the poor results achieved by their NERC in *asrC*. Thus, their accuracy and MRR are not comparable measures.

Regarding factoid questions, SIBYL's baseline are behind the ones of LIMSI by more than 8 points in MRR and accuracy on the manual transcripts. INAOE has also a much larger MRR score. Top5 shows that our baseline system has a low coverage, 10 answers behind INAOE (25% less) and 6 behind LIMSI. The results for question set B that we report in TAble 2.8 are much better than the official ones [Turmo et al., 2009] due to the posterior correction of an important bug in the question classification module. We report these updated results because the comparison using the original ones would be meaningless.

We can see that moving from manual transcripts to *asrA* transcripts has very little impact in SIBYL's MRR and accuracy. Subsequent increase of WER by 13.5% on *asrC* transcripts has no additional impact (results are slightly better due to the correct recognition of one additional “nil” question). Only on *asrB* transcripts our results are considerable worse than in the other transcripts. Although *asrB* WER (14.0%) is much lower than on *asrC* and almost the same as *asrA*, these transcripts achieve the worst results in MRR and accuracy for any of the evaluated systems.

Although our initial results on manual transcripts are worse than LIMSI and INAOE, the degradation produced by ASR transcripts is much lower. SIBYL has better MRR than INAOE and LIMSI on *asrB* and *asrC* respectively, being more robust when dealing with ASR transcripts. LIMSI lose 12 points of MRR when moving from manual to *asrC* transcripts and INAOE losses almost 9.

We have taken the 2009 English EPPS task as the definitive testbed for our system, es-

			Factoid			Definitional			All	
Trans.	**System**	**Q.Set**	**Top5**	**MRR**	**Acc.**	**Top5**	**MRR**	**Acc.**	**MRR**	**Acc.**
	INAOE1	A	35	0.32	24.0%	6	0.21	20.0%	0.30	23.0%
		B	34	0.33	25.3%	6	0.21	20.0%	0.30	24.0%
	INAOE2	A	35	0.32	22.7%	7	0.22	20.0%	0.29	22.0%
		B	34	0.32	24.0%	7	0.22	20.0%	0.29	23.0%
asrA	LIMSI1	A	32	0.34	28.0%	10	0.25	20.0%	**0.31**	**26.0%**
		B	30	0.31	25.3%	11	0.29	24.0%	0.30	25.0%
10.6%	TOK1	A	13	0.08	4.0%	3	0.04	0.0%	0.07	3.0%
		B	12	0.07	2.7%	4	0.08	4.0%	0.07	3.0%
	UPC1	A	29	0.27	18.7%	7	0.26	24.0%	0.27	20.0%
		B	26	0.25	17.5%	7	0.24	26.9%	0.25	20.0%
	UPC2	A	30	0.26	18.7%	6	0.24	24.0%	0.26	20.0%
		B	27	0.22	13.5%	8	0.28	26.9%	0.24	17.0%
	INAOE1	A	23	0.22	16.0%	6	0.21	20.0%	0.22	17.0%
		B	23	0.21	13.3%	7	0.25	24.0%	0.22	16.0%
	INAOE2	A	24	0.22	16.0%	6	0.21	20.0%	0.22	17.0%
		B	24	0.21	13.3%	7	0.25	24.0%	0.22	16.0%
asrB	LIMSI1	A	24	0.27	22.7%	8	0.20	16.0%	0.25	**21.0%**
		B	24	0.26	21.3%	9	0.24	20.0%	0.25	21.0%
14.0%	TOK1	A	9	0.06	4.0%	3	0.03	0.0%	0.06	3.0%
		B	10	0.06	2.7%	3	0.06	4.0%	0.06	3.0%
	UPC1	A	26	0.24	17.3%	7	0.26	24.0%	0.24	19.0%
		B	21	0.19	13.5%	8	0.28	26.9%	0.21	17.0%
	UPC2	A	29	0.26	20.0%	7	0.25	24.0%	**0.26**	**21.0%**
		B	25	0.21	14.8%	8	0.28	26.9%	0.23	18.0%
	INAOE1	A	29	0.31	26.7%	5	0.20	20.0%	**0.28**	**25.0%**
		B	28	0.30	26.7%	5	0.20	20.0%	0.28	25.0%
	INAOE2	A	29	0.30	25.3%	6	0.21	20.0%	**0.28**	24.0%
		B	28	0.29	24.0%	6	0.21	20.0%	0.27	23.0%
asrC	LIMSI1	A	23	0.26	24.0%	8	0.19	12.0%	0.24	21.0%
		B	24	0.24	21.3%	9	0.23	16.0%	0.24	20.0%
24.1%	TOK1	A	17	0.12	5.3%	5	0.08	4.0%	0.11	5.0%
		B	19	0.11	4.0%	5	0.12	8.0%	0.11	5.0%
	UPC1	A	22	0.21	16.0%	6	0.24	24.0%	0.22	18.0%
		B	24	0.23	17.6%	7	0.28	26.9%	0.24	20.0%
	UPC2	A	26	0.24	17.3%	6	0.24	24.0%	0.24	19.0%
		B	26	0.23	16.2%	8	0.28	26.9%	0.25	19.0%

Table 2.9: Results of the 2009 task T1: English EPPS automatic transcripts (75 factoid questions and 25 definitional ones)

pecially for our experiments in answer extraction (Chapter 7) and coreference (Chapter 8). These extensions of Sibyl were developed after the CLEF 2009, when we performed a major rewriting and bug-correction of Sibyl's source code. This is the reason because the results for written questions (set A) are slightly different from the official ones of [Turmo et al., 2009]. Additionally, this updated version of Sibyl will be used as a baseline for further improvements in the next chapters.

2.4 QA Technology applied to Speech Transcripts

Recent research projects, like RITEL[11] and the European Commission funded QALL-ME,[12] have aimed to create infrastructures for open domain Question Answering through telephone. Voice interfaces to QA (or *spoken QA*, or *voice-activated QA*), although related to speech technologies, use speech only in their human-machine interface in a interactive communication [Harabagiu et al., 2002, Stenchikova et al., 2006, van Schooten et al., 2007]. In this situation, due to the interactivity of the task, the user can reformulate the questions as many times as necessary until the computer correctly recognises the speech [Cabrio et al., 2008] or until an acceptable answer is achieved. The techniques of spoken QA are related to dialogue and automatic speech recognition but essentially orthogonal to the QA on speech transcripts issues. The spoken documents mostly have only one audio recording, and no possibility to clarify the utterances in case of transcript errors. Additionally, long speech utterances have looser grammar than questions (as we can see from the QAst 2009 data-sets), where the user must express what they want with high precision. Thus, it is not straightforward how the experience on voice-activated QA can be relevant about how to deal with spoken documents.

Regarding the problem of QA on spoken documents, there is very little literature about it. Outside our first preliminary work on this topic [Surdeanu et al., 2006], we are aware of only one proposal tackling this task before the beginning of the QAst evaluations in CLEF 2007. The work of Akiba and Tsujimura [2007] aims at building an error-tolerant question answering system for factoid questions on spoken documents. The output of this QA systems is not concrete answers but full utterances (i.e., a section of speech), thus, being an utterance retrieval engine with no Answer Extraction module. They propose to use a set of ML-based classifiers that detect if an utterance contains a named entity, although they do not extract the exact entity. As a consequence, several types of named entity can be simultaneously detected in the same utterance. This method makes the system more robust to recognition errors than a rule-based named entity detector. The named entity classifiers are SVMs and the features are $n-$grams of words and POS, there is one different classifier for each entity type. The utterances that contain a named entity

[11] http://ritel.limsi.fr

[12] http://qallme.fbk.eu

of the expected answer type are ranked according to a combination of two scores: one is the confidence of the named entity detector and the other is the likelihood obtained from the IR for this utterance. They test the performance with a set of ASR transcripts of a television show in Japanese. The reported results are an MRR of 0.404 for the IR baseline, and a MRR of 0.441 for the model combining IR and named entity detection.

The QAst tracks are the only evaluation framework for the spoken document QA up to this day. In the QAst evaluation, it is compulsory to provide the exact answer string (or time interval) that corresponds to the entity the question asks for, thus, the QAst systems follow more closely the architecture we have described in Section 2.2.1.

2.4.1 The QAst Systems

In the rest of this section, we sketch the characteristics of all systems that have been evaluated in QAst. We do not comment on the characteristics of our own participations with the UPC team, since the presented systems are a baseline of the full SIBYL system that is presented through the rest of this dissertation.

Table 2.10 lists the speech QA systems in QAst and summarises their characteristics broken-down to its basic modules. The characteristics are divided in five groups: 1) linguistic information used (Enrichment), 2) question processing strategies, 3) passage retrieval strategies, 4) answer extraction for factoid questions, and 5) named entity detection and classification (NERC). All the systems shown in Table 2.10 work with speech transcripts (either manual or automatic) and do not include an integrated ASR subsystem. As described before, spoken language is different from written text in numerous ways and ASRs introduce unexpected errors in the transcripts. These QA systems use only very shallow and robust natural language information and rely more on heuristic and statistical information than on linguistic knowledge. Regarding their characteristics:

1. All systems enrich the input text with some linguistic information: named entity detection is almost compulsory for factoid QA. Part-of-speech tagging and lemmatisation are helpful to form more general patterns than word forms alone, avoiding sparsity of the lexical items.

2. Regarding Question Classification, it is general QA problem. Because the question is given in the form of written text, Question Classification is an equivalent problem for QA on spoken documents and on written text. Given this and that no error analysis is provided by the authors of the described systems, Question Classification will not be further discussed here.

3. For Passage Retrieval, all systems apply some kind of traditional IR with standard ranking functions. They either use popular IR engines (e.g., Indri, Lucene) or in-house ones (e.g., the UA system uses IR-n [Pardiño et al., 2008]). The LIMSI system [Bernard et al., 2009] uses a weighted selection of question features called *descriptors*, and INAOE [Reyes-Barragán et al., 2009] uses the popular Soundex algorithm to convert documents and question words into sound codes that help improve retrieval coverage on automatic transcripts.

4. For Answer Extraction, all systems use very shallow measures of relevance. Frequent they use the average distance between the candidate answer and the keywords found in the passage (CUT, LIMSI, UA). Other rankers use the retrieval's confidence score or some form of candidate redundancy (CLT, DFKI, INAOE). All the reranking functions are robust heuristics suitable for both text and speech, but they have a poor performance compared to more in-depth processing methods that use sophisticated linguistic knowledge. The LIMSI system [Bernard et al., 2009] has two different extensions of these heuristic measures. The first uses a Bayesian modelling of the context of a correct answer in terms of distance and redundancy. The second one is a reranking method that scores the answer according to a distance measure between the question and the answer context in terms of tree similarity. These trees are obtained by hierarchically grouping syntactic chunks using labelled relations and handcrafted rules. A set of transformation operations, with empirically set costs, is also defined. Unfortunately, this second method has only been applied to French QA, and the results cannot be compared to the rest of English systems.
5. Regarding NERC, almost all systems use handcrafted rules. CLT [Mollá et al., 2007] uses a NERC module specific for automatic speech transcripts. It is based on machine learning (maximum entropy models), and combines handcrafted regular expressions with gazetteers of named entities and contextual features (e.g., capitalisation, presence of digits and prepositions). The system is trained using a mix of the BBN corpus[13] (news) and the AMI corpus of spontaneous speech transcripts (multi-part meetings).

Several of these systems are written text QA systems adapted for this task: DFKI [Neumann and Wang, 2007] uses an existing QA system for written text, including a statistical NERC trained for text, and CUT [Kürsten et al., 2008] uses off-the-shelf linguistic processors to identify candidate named entities from passages. A series of rules match expected answer types with named entities. And these are ranked according to their average distance to the keywords when some set confidence thresholds are met. CLT uses a written text QA system adapted for speech. Due to the disfluencies of spontaneous speech, they do not use their usual algorithms for syntactic and graph-semantic information [Van Zaanen et al., 2006]. Their question answering strategy is entirely based on finding and selecting the right named entities.

The TOKIO system [Heie et al., 2009] is a special case. This is a completely data-driven statistical QA system that makes no use of linguistic information. It is mainly a language modelling engine: given a question Q, a document D is ranked according to the conditional probability of generating Q, $P(Q|D)$. Then the Answer Extraction module models the probability of an answer A given Q as $P(A|Q) = P(A|W, X)$, where W are the features describing the question type and X question words. All these conditional probability distributions are learnt from the QAst development questions.

All of the participants in the three QAst evaluations (except TOKIO) used named entity recognition to generate candidate answers, and very superficial measures for answer extraction (what we have classified as simple lexical *answerness* measures in Section 2.2.2).

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

Name	**Enrichment**	**Question Process**	**Passage Retrieval**	**Factoid Answer Extraction**	**NERC**
CLT	words and NEs	hand-crafted rules	passage ranking based on word similarity between passage and query	candidate ranking based on frequency and NERC confidence	hand-crafted patterns, gazetteers and ME modules
CUT	words, NEs and POS	hand-crafted rules	passage ranking based on retrieval score	candidate ranking based on keyword distance and retrieval score	Stanford NERC, rules with classification
DFKI	words and NEs	hand-crafted syntactic-semantic rules	Lucene IR engine	candidate ranking based on frequency	gazetteers, statistical models tuned for writen text
INAOE	words, NEs and phonetics	hand-crafted rules	Indri IR engine	candidates sorted by passage retrieval confidence score	regular expressions
LIMSI	words, lemmas, named entities, morphologic derivations and synonymic relations	hand-crafted rules	passage ranking based on search descriptors	ranking based on either keyword distance and redundancy, Bayesian modelling or tree distance	hand-crafted rules with statistical POS tags
TOKIO	words and word classes derived from training data, question–answer pairs	—	sentence ranking based on statistical models	ranking based on analogy between input question and answers on the training data	—
UA	words, NEs, POS and n-grams	hand-crafted rules	ranking based on n-grams	ranking based on keyword distance and mutual information	hand-crafted rules

Table 2.10: Synoptic table of systems evaluated in QAst

No system used more elaborated linguistic answer extraction methods, like the ones dropped by CLT, than measuring redundancy, distance and IR scores. Thus, not allowing us to compare the performance of these methods with respect standard written text. The syntactic tree similarity method used by LIMSI in the French manual transcripts was the only system to use a measure of similarity between question and candidate answer. However, it did not yield any improvement over the baseline [Bernard et al., 2009].

3. The Sibyl Question Answering System

This Chapter is an overview of Sibyl, a full QA system designed to work with spoken documents, and serves as an index for the rest of this dissertation.

Sibyl's main three modules strictly follow the architecture from Figure 2.1 that is common for most of the QA systems (Section 2.2.1. The following five Chapters describe in detail how Sibyl's modules work and make a white-box evaluation of their individual performance, including the preprocess and enrichment related modules of named entity recognition and coreference resolution:

- Chapter 4 describes Sibyl's Question Processing module,
- Chapter 5 describes Sibyl's Passage Retrieval module,
- Chapter 7 describes Sibyl's Answer Extraction module,
- Chapter 6 describes Sibyl's NERC module, finall, Chapter 8 describes Sibyl's Coreference Resolution module.

3.1 Overview of Sibyl

Figure 3.1 shows all the modules in Sibyl and how they are related. The leftmost column contains Sibyl's three main modules, the central column shows the tools and knowledge sources used by the QA modules, the rightmost column shows the information inputs that must be provided by the user in order to obtain a ranked answers list.

Input: Sibyl is provided with a written question as main input. This question is processed with NERC and dependency parsing.

Preprocess: In a previous stage, the collection of speech transcripts is processed and collated into an index. Automatic speech recognition is not built in Sibyl. This processing includes part-of-speech tagging, lemmatisation, named entity recognition (NERC) and dependency parsing. These tools are the same used to process the input question. Additionally, coreference resolution is available, although we have used it only for manual transcripts. Finally all this information is stored in a searchable index.

Question Classification: The first of Sibyl's modules processes the input question and produces two outputs: the expected answer type of this question (EAT) as a list of the most probable EATs, and also a list of selected keywords from the question text.

Passage Retrieval: The Passage Retrieval module takes the keyword list and searches relevant passages in the collection. This retriever dynamically selects what keywords to use and the length of the passages depending on the documents. Special features for dealing with automatic transcripts can be activated in this module (the PHAST SDR engine), also coreference information can be used.

Answer Extractor: This module takes the set of passages retrieved from the previous module and the EAT list. It outputs a sorted list of candidate answers of the EAT type extracted from the passages. Internally, it is a two step process:

1. Candidates are selected from the named entities matching the EAT, and a set of heuristic scores is calculated for each one.
2. A machine learning reranker takes these candidates and reranks them using the previous scores and other information such as a measure of syntactic similarity between questions and passages. The reranking model has been previously learnt from a set of development questions.

Rerank Learner: This module learns a candidate scorer given a set of passages, candidate answers, and the correct answers. This is a binary classifier, each candidate is considered either a correct or incorrect answer.

Output: The final output of the system is a ranked list of answers.

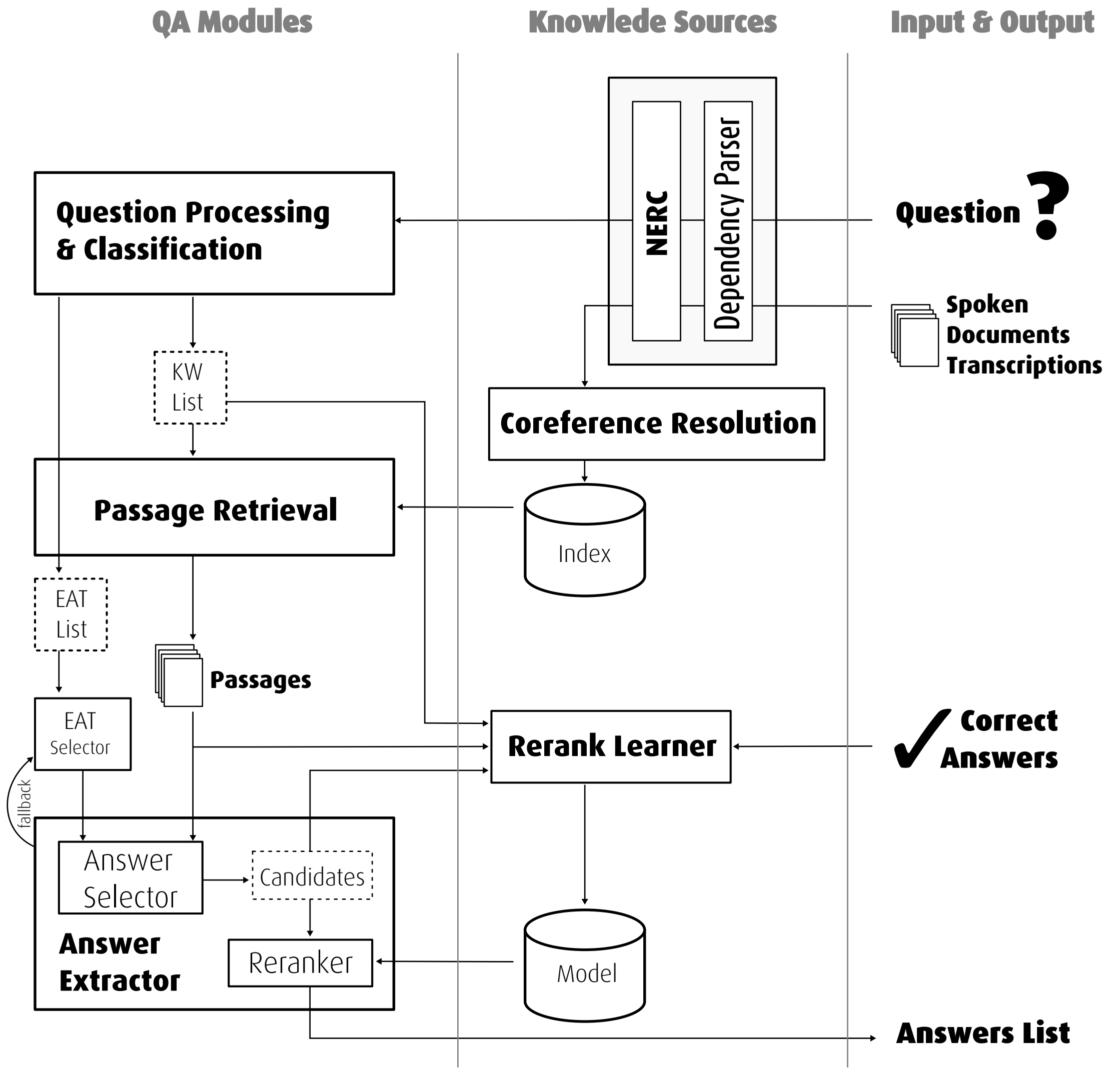


Figure 3.1: Architecture of the Sibyl Question Answering System: modules and information flow between them

3.2 Implementation of Sibyl

Sibyl has been implemented using the metaserver framework [Gonzàlez, 2009]. This is a generic client/server architecture designed to implement interactive text-mining applications. It aims to provide a framework where processing modules can be easily reused accross applications and easily replaced within applications. Sibyl is divided in several metaserver modules that share information using a predetermined file format (mainly column-based plain text with metadata) that the metaserver API can easily handle. The metaserver API is available for several programming languages: `C/C++`, `Perl`, `Java` and `Python`. Most of sibyl is implemented in `Java`, the PHAST SDR engine is written in `C++` for the sake of efficiency, and many parts of the learning/reranking modules are glued with `Perl`.

The tools used to preprocess the documents collection are part of our in-house annotation facilities, capable of processing large volumes of text in an computers cluster running Oracle Grid Engine.

Regarding the time performance of sibyl, she can process the 100 test questions of QAst 2009 in ten minutes in a single CPU machine. Although this performance is achieved when both documents and questions have previously been processed with our linguistic annotators, which is unlikely out of this controlled laboratory experiments.

4. Question Processing and Classification

This Chapter describes the question processor module of Sibyl. The main goal of this component is to detect the expected answer type (sometimes referred as EAT) for the current question. For factoid QA, this means determining what types of named entities can or cannot be the answer. Question Classification is a general QA problem since it involves only the question but not the documents, it is not specifically related to the QA on spoken documents scenario. The research supporting this thesis has not explicitly addressed the question classification problem in the context of spoken documents, we have implemented a classifier comparable with state-of-the-art results using well-known techniques.

Additionally to the classification task, this module analyses the question and extracts from it all the information that later will be necessary for the rest of the QA process. The exact nature of this task depends on what strategy for Passage Retrieval and Answer Extracted will be used. In Sibyl, the processing involves selecting a set of query terms from the question words and using a dependency parser to get the syntactic structure.

This Chapter explains Sibyl's strategy for Question Processing and Classification, and discusses its performance.

ABBREVIATION:abb	ENTITY:other	LOCATION:mount
ABBREVIATION:exp	ENTITY:plant	LOCATION:other
DESCRIPTION:def	ENTITY:product	LOCATION:state
DESCRIPTION:desc	ENTITY:religion	NUMBER:code
DESCRIPTION:manner	ENTITY:sport	NUMBER:count
DESCRIPTION:reason	ENTITY:substance	NUMBER:date
ENTITY:animal	ENTITY:symbol	NUMBER:distance
ENTITY:body	ENTITY:techmeth	NUMBER:money
ENTITY:color	ENTITY:termeq	NUMBER:order
ENTITY:cremat	ENTITY:veh	NUMBER:other
ENTITY:currency	ENTITY:word	NUMBER:perc
ENTITY:dismed	HUMAN:description	NUMBER:period
ENTITY:event	HUMAN:group	NUMBER:speed
ENTITY:food	HUMAN:individual	NUMBER:temp
ENTITY:instrument	HUMAN:title	NUMBER:volsize
ENTITY:lang	LOCATION:city	NUMBER:weight
ENTITY:letter	LOCATION:country	

Table 4.1: List of Expected Answer Type labels with the structure of TYPE:subtype

4.1 Expected Answer Type

Sibyl recognises the 50 open-domain answer types defined by Roth [Li and Roth, 2005].[1] Table 4.1 shows these answer types. 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. The classifier obtains an accuracy of 87.6% on the corpus of Roth. Then, these 50 different answer types are mapped into a set of named entity types suitable for the task of factoid QA or are discarded if they are definitional questions. Since NERC systems usually recognise less than ten different entities, the 50 answer types are basically grouped into 10 general classes.

For example, questions with location-related answer types are mapped as follows: questions regarding *countries*, *states* or *mountains* will take 'location' entities as candidates answers, but questions about *cities* or *other* locations will be answered with either 'location' or 'organisation' entities. Many questions are inherently ambiguous between 'location' and 'organisation' types. This mapping helps increasing the recall of candidate answer detection. As an example of this ambiguity, consider QAst question *Where is Mister Buttiglione from?*. Our classifier assigns it an expected *Location:Other* answer type. This question is ambiguous because the user has under-specified what kind of place should

[1] This is a frequently used resource. This data collection can be downloaded from the authors `http://cogcomp.cs.illinois.edu/Data/QA/QC/`

Salience	Type of Word
9	words within quotes
8	named entities
7	sequences of nouns and adjectives
6	sequences of nouns
5	adjectives
4	nouns
3	verbs and adverbs
2	question focus word
1	any non-stop word

Table 4.2: Keywords and their salience

Mr. Buttiglione come from. He could either come from a physical location or have been sent by some other entity like the corporation he works for. You can not assure what is the desired answer even looking at information in the documents.

This answer type predictor can be adapted to other types and other languages, for example Spanish as seen in our work [Comas and Turmo, 2008b].

4.2 Keywords Selection

The Question Processing component selects a set of relevant keywords from the question to be used in as queries for the Passage Retriever. First, the question is enriched with part-of-speech tagging and named entity classification, and the question focus word (i.e. the most unlikely question word to appear in the answer) is detected using a set of syntactic rules. Then the keywords are selected with an heuristic process. Each keyword is assigned an heuristic salience value depending on the part-of-speech and name entity information, the higher the value, the more important the keyword is. Keywords are assigned the highest possible value from those shown in Table 4.2, stop words and question tags are ignored. This salience information is later used in the Passage Retrieval module to issue queries to the IR engine (see Section 5 for details).

For example, consider this question from the QAst 2009 English task: *How many countries are member of the U.N. Security Council?* Our Question Processor assigns the following salience values

Word	POS	Salience
u.n.	NNP	8
security	NNP	8
council	NNP	8
member	NN	4
countries	NNS	2

Year	Task	non-nil factoid q.	Accuracy
2009	T1a: English EPPS	64	76.5%
2009	T1b: English EPPS Spont.	64	73.4%
2009	T2a: Spanish EPPS	49	79.5%
2009	T2b: Spanish EPPS Spont.	49	55.1%
2008	T1: English CHIL	73	89.0%
2008	T3: English AMI	70	78.5%
2008	T4: English EPPS	73	80.8%
2008	T5: Spanish EPPS	69	26.0%

Table 4.3: Accuracy of the Expected Answer Type predictor in the 2008 and 2009 QAst evaluations

after identifying *U.N. Security Council* as a named entity and *countries* as question focus word. The Passage Retrieval module will consider that *countries* has little relevance when retrieving the documents —since is expected that these documents will contain the actual name of countries rather than the word *countries* In contrast, the named entities will be a priority for the retriever, as they are expected to appear close to the answer in the text.

Named entities are detected with our in-house NERC described in Section 6, the TnT tagger is used for part-of-speech tagging [Brants, 2000].

4.3 Evaluation and Discussion

We have evaluated the performance of Sibyl's Expected Answer Type predictor in the 2008 and 2009 QAst evaluations. We have discarded any definitional question and have focused on the accuracy of the factoid questions prediction. The results are shown in Table 4.3.

For the English tasks, the EAT predictor has have an average accuracy of 79%, with lower values in QAst 2009 tasks. Since the predictor model has been learnt from a corpus of written questions, it is expected that spoken questions are more difficult to classify. The difference when using spontaneous questions or corrected questions is just a 3% for English, showing this is a minor issue here. In task T5 from QAst 2008, the accuracy was only 26%, this poor performance is due to methodological mistakes occurred when processing the questions. Those were fixed later. The results of Spanish task are better than English in QAst 2009 but there is a loss of 24 points when using the spontaneous questions. This shows that our tools for processing Spanish are less robust than for English.

5. Passage Retrieval

This component retrieves a set of relevant passages from the document collection, given the keywords previously extracted from the question. This Chapter presents our approach to information retrieval for spontaneous speech corpora. The classical approach to this problem is the use of an automatic speech recogniser (ASR) combined with standard information retrieval techniques. However, ASRs produce transcripts with significant word error rate, which is a drawback for standard retrieval techniques. We present two versions of the retriever, the former is a written-text based passage retriever and the latter is designed for ASR transcripts. This is based on an approximated sequence alignment algorithm to search "sounds like" sequences.

The main content of this Chapter has been previously published in the conference paper [Comas and Turmo, 2008a].

Query Keywords: relevant, documents, process

Passage:

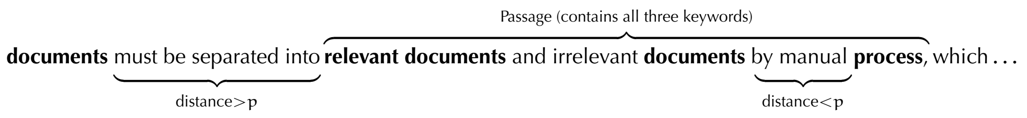

Passage (contains all three keywords)
documents must be separated into **relevant documents** and irrelevant **documents** by manual **process**, which . . .
distance>p
distance<p

Figure 5.1: Example of passage building using DQ algorithm

5.1 Written Document Retrieval

Empirical studies [Pasca, 2001, Surdeanu et al., 2006] show that better results in QA are achieved when using a retrieval method based in passage-building procedures rather than document ranking measures. For this method, first the question is processed to obtain a list of keywords ranked according a linguistically motivated priority. Then some of the most salient keywords are sent to the IR engine as a Boolean query. A word distance threshold p is also set in order to produce passages of high keyword density. All documents containing those passages are returned as an unordered set. If this set is too large or small, keywords and p may be altered iteratively. The rationale behind this algorithm is to produce compact passages that contain as much query words as possible in a small span of text. The benefits produced by this kind of IR strategy for QA is supported by related research [Monz, 2004].

Sibyl's Passage Retrieval algorithm uses a dynamic query (DQ) adjusting procedure, which iteratively adjusts the number of keywords used for retrieval and their proximity until the quality of the recovered information is satisfactory. In each of these iterations, a Document Retrieval application fetches documents relevant to the current query, and a subsequent passage construction module builds passages as segments where two consecutive keyword occurrences are separated by at most p words. We have used the Lucene IR engine[1] to implement this module.

The DQ algorithm is shown in Algorithm 1. The initial parameters have been set experimentally: the set of keywords K is initialised to all keywords with a salience of 2 or more, and the current proximity is initialised to 50 words. The algorithm is configured with four parameters:

- $MinPass$ and $MaxPass$: lower and upper bounds for the acceptable number of passages (currently set to 1 and 50).
- $MinProx$ and $MaxProx$: lower and upper bounds for keyword proximity (currently set to 50 and 100 words).

[1] http://lucene.apache.org/

Algorithm 1: DQ Algorithm

```
Input: keyword set $\mathcal{K}$, maximum word proximity $p$
retrieve passages using current $\mathcal{K}$ and $p$;
while number of passages<MinPass or number of passages>MaxPass do
    if number of passages<MinPass then
        if $p$<MaxProx then
            increase $p$;
        else
            reset $p$;
            drop the least-significant keyword from $\mathcal{K}$;
        end
    else if number of passages > MaxPass then
        if $p$>MinProx then
            decrease $p$;
        else
            reset $p$;
            add the next available keyword to $\mathcal{K}$;
        end
    end
end
return current set of passages;
```

This algorithm uses very limited linguistic information (only part-of-speech tags for keyword ranking in the question), which makes it very robust for speech transcripts. Figure 5.1 shows an example of passage construction for a simple query and one sample sentence. In this example a passage containing all three keywords is found, being each pair of subsequent keyword occurrences within a distance of p or less words. The first occurrence of “documents” is too far from “relevant” to be included in the passage.

5.2 Spoken Document Retrieval

Classically, the approach to the spoken document retrieval (SDR) problem is the integration of an automatic speech recogniser with 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.

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, and then apply a wide variety of IR techniques [Alzghool and Inkpen, 2006, Jones et al., 2006, Inkpen et al., 2006a, Wang and Oard, 2005]. Other methods take advantage of internal information supplied by the ASR such as confidence scores, n-best output and full lattices. This information may be used to improve the retrieval performance in several ways [Srinivasan and Petkovic, 2000, Saraclar and Sproat, 2004] but makes the system dependent on a concrete ASR.

There is a vast literature on SDR for non-spontaneous speech. For example, the TREC conference had a spoken document retrieval task using a corpus of Broadcast News. The TREC 2000 edition concluded that spoken news retrieval systems achieved almost the same performance as traditional IR systems [Garofolo et al., 2000]. Spontaneous speech contains disfluencies that can barely be found in broadcast news, such as repetition of words, use of onomatopoeias, mumbling, long hesitations and simultaneous speaking. The Spoken Document Retrieval track in the CLEF evaluation campaign uses a corpus of recorded interviews for the task of cross-lingual speech retrieval in spontaneous speech (CL-SR) [Oard et al., 2006, Pecina et al., 2007]. This is a more general scenario than former TREC tracks, most of the work done by the participants is focused on investigating the effects of meta-data, hand-assigned topics, query expansion, thesauri, side collections and translation issues.

Some researchers have 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 [Inkpen et al., 2006b] showing only slight improvements.

In this Section we present an IR engine designed to tackle ASR recognition errors by specifically recognising erroneously transcribed words. This approach is totally ASR independent and uses only the 1-best transcript.

5.2.1 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 signals and phones, and a *language model*, which estimates the probability of a certain sequence of words. Given the input signal A, a word sequence W' is generated according to the following rule:

$$W' = \arg\max_{b} P(A|W_i)P(W_i) \tag{5.1}$$

where W' is the most likely word sequence, having the maximum *a posteriori* probability in the model. $P(A|W_i)$ is the probability of word sequence W_i sounding like A (acoustic model) and $P(W_i)$ is the probability of the sequence W_i occurring in the language (language model).

Any ASR system is limited in the size of the vocabulary it can recognise depending on the amount of data used when learning the language model. The words it cannot recognise are called Out Of Vocabulary (OOV) words. ASR transcribes the audio corresponding to these words as other sequence of words from its closed vocabulary that could produce a similar sound. Words such as proper names tend to be OOV. When dealing with automatic transcripts, the incorrectly transcribed words may create a word recognition problem for the IR engine, introducing false positives and false negatives to its input.

5.2.2 Phonetic Alignment Search Tool

To deal with such difficulties, we have implemented an SDR engine relying on phonetic similarity for the automatic transcripts. This tool is called PHAST (PHonetic Alignment Search Tool), and uses approximated pattern-matching algorithms to search for small sequences of phones (the keywords) in a larger sequence (the documents) using a measure of sound similarity. Thus, it can identify the keywords and also groups of words that "sound like" the searched keywords. This procedure relies on the hypothesis that most of the OOV words have been transcribed into phonetically related vocabulary words.

We have implemented PHAST (Algorithm 2) following the approach of BLAST [Altschul et al., 1990]. BLAST is a pattern-matching algorithm used in biological sequences to identify protein homology (i.e., searching sequences in large DNA databases). Both algorithms operate under the same hypothesis: it is possible to find the best matchings of the keywords by searching for small, contiguous substrings of phones (called *hooks*) in the transcript, extending them, and computing their relevance. Both algorithms aim to perform computationally cheap and fast searches in large databases.

PHAST algorithm has some advantageous properties for dealing with SDR: it finds approximated matchies independent of sub-word length, it can easily split/merge sequences, and no training data is required. This process operates on sequences of phones, making no language-related assumptions during the search process. Algorithm 2 shows a general view of PHAST. 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 phonetically transcribed keywords ($\mathcal{KW}$). This is the only language-dependent part of the whole process. To perform this transcription, we have used the Carnegie Mellon Pronouncing Dictionary[2] and the `SoundChange` package from `Perl`'s library as a grapheme-to-phoneme mechanism for unknown words.

Algorithm 2 has three important functions that are described in detail below:

- $\mathtt{detection}_\phi(w, \mathrm{d})$: This function detects the position of hooks h within document d considering keyword w and using the search function ϕ. Similarly to Altschul et al. [1990], function ϕ has been implemented as follows. Given a set of phonetically transcribed keywords, a deterministic finite automaton DFA_k is automatically built for each keyword k in order to recognise all its possible substrings of n phones.

[2] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

Algorithm 2: PHAST Algorithm

```
Input: $\mathcal{D}$: collection of phonetically transcribed documents
$\mathcal{KW}$: set of phonetically transcribed keywords
forall the $d \in \mathcal{D}, w \in \mathcal{KW}$ do
    while $h = \mathtt{detection}_{\phi}(w, d)$ do
        $s = \mathtt{extension}_{\varphi}(w, h)$;
        if $\mathtt{relevant}(s, h)$ then
            update $\mathtt{tf}(w, d)$;
        end
    end
end
Rank collection $\mathcal{D}$;
```

For instance, given $n = 3$ and the keyword "alignment," which is phonetically transcribed as [əlaɪnmɪnt],[3] there are seven phone substrings of length three (3-grams): əla, laɪ, aɪn, ɪnm, nmɪ, mɪn and ɪnt. One DFA is automatically built to recognise all seven 3-grams at once. Using these DFAs, the collection is scanned once to search for all the hooks.

- $\mathtt{extension}_{\varphi}(w, d, h)$: After a hook h is found, PHAST uses φ to extend it in document d and to compute its score value s. Function φ has been implemented with a phonetic similarity measure due to the success achieved in other research domains [Kessler, 2005]. Specifically, we have used a flexible and mathematically sound approach to phonetic similarity proposed by Kondrak [2000]. This approach computes the similarity $\Delta(a, b)$ between two phone sequences a and b using the edit distance implemented with a dynamic programming algorithm. This implementation includes two new operations of compression and expansion that allow the matching of two contiguous phones of one string to a single phone from the other. (e.g., [c] sounds like the pair [tʃ] rather than [t] or [ʃ] alone).

  The score value s is finally computed by normalising the similarity $\Delta(a, b)$ by the length of the matched substrings $\mathtt{length}(a, b)$:

$$s = \frac{\Delta(a, b)}{\frac{\Delta(a, a)}{n} \cdot \mathtt{length}(a, b)}, \tag{5.2}$$

  where n is the length of the longest string, either a or b.

- $\mathtt{relevant}(s, h)$: This judges whether the occurrence of w at h with score s is relevant enough for term frequency. An occurrence is relevant when its score is larger than a given threshold t. In Algorithm 2, tf is updated when w is a relevant occurrence. Initial experiments have shown that, on the one hand, the best results are achieved when low scoring matches are filtered out, and on the other hand, the best results

[3] We have used the international phonetic alphabet (IPA) notation for phonetic transcriptions.

Reference transcript T1: *The host system it is a UNIX Sun workstation*
Automatic transcript T2: *that of system it is a unique set some workstation*

	jun	$\leftarrow \mathtt{detection}_\phi$	
...ðæt ʌβ sɪstəm ɪt ɪz ə	jun	ik sɛt sʌm	wəʊrksteɪʃən...
	junik s	sʌn	$\leftarrow \mathtt{extension}_\varphi$

Figure 5.2: Search of the term "UNIX-Sun" in the QAst 2007 T2 corpus

are achieved with $tf \leftarrow tf + s$ rather than $tf \leftarrow tf + 1$. This helps to filter out false positives, especially for very common syllables.

Figure 5.2 shows an example of how the $\mathtt{detection}_\phi$ and $\mathtt{extension}_\varphi$ functions are used. Document d is the sentence *"The host system it is a UNIX Sun workstation",* which has been transcribed into 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 *M* but not in the erroneous automatic transcript *asrA*. In the first step, $\mathtt{detection}_\phi$ finds the hook [jun] related to [juniks sʌn]. In the second step, $\mathtt{extension}_\varphi$ extends the hook by matching the rest of [juniks sʌn] with the phones surrounding [jun] in the sentence.

As stated before, the $\mathtt{extension}_\varphi()$ function computes the normalised cost of the edit distance between two phoneme sequences $a = a_i a_2 \ldots a_n$ and $b = b_1 b_2 \ldots b_m$. This cost is a function of similarity, not of distance, and does not have the mathematical properties of a distance measure.

The edit distance function $\Delta(a, b)$ uses a measure of inter-phoneme similarity between pairs of phonemes $\delta(a_i, b_j)$ which is based on the phone features described by Kondrak [2002]. Each phoneme is described from a physical point of view with multivaluated features (e.g. articulatory point, roundness). As an example, Table 5.1 shows the features used for consonants; the number enclosed in parentheses is the numerical value of the feature. Table 5.2 shows the salience of each feature. These values are used to define a similarity value $\delta(a_i, b_j)$ for each phoneme and each possible edit operation (i.e. substitution, insertion/deletion, compression/expansion):[4]

$$\begin{aligned}
\delta(a_i, empty) &= k_1 && (5.3)\\
\delta(a_i, b_j) &= k_2 - \delta'(a_i, b_j) - V(a_i) - V(b_j) && (5.4)\\
\delta(a_i a_{i+1}, b_j) &= k_3 - \delta'(a_i, b_j) - \delta'(a_{i+1}, b_j) - && (5.5)\\
&\quad V(b_j) - max(V(a_i), V(a_{i+1})) && (5.6)
\end{aligned}$$

where k_1 is the cost of deleting a symbol, k_2 is the base score when matching two equal symbols and k_3 is the base score when compressing two symbols into one. $V(a_i)$ and $\delta'(a_i, b_j)$ are defined as:

[4] The $\delta(a, b)$ function is symmetric. For the sake of simplicity, just one direction is presented.

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

Table 5.1: Features for consonants. V, N, R, L, and T stand for *Voice*, *Nasal*, *Retroflex*, *Lateral*, and *Trill*

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 5.2: Phone features and their salience

$$V(a_i) = \begin{cases} 0 & \text{if } a_i \text{ is a consonant} \\ k_4 & \text{otherwise} \end{cases} \tag{5.7}$$

$$\delta'(a_i, b_j) = \sum_{f \in features} \mathtt{diff}(a_i, b_j, f) \cdot salience(f) \tag{5.8}$$

where k_4 is a penalty cost for matching vowels with consonants and $\mathtt{diff}(a, b, f)$ evaluates whether feature f is different in a_i and b_j. The values of the constants k_i have been set heuristically: k_1 =-1000, k_2 =3500, k_3 =3500 and k_4 =1000 following Kondrak [2002].

In the last line of the PHAST algorithm, the term frequencies $\mathtt{tf}$ found in the text can be used to rank the documents with any standard relevance measure used in IR. In particular, they can also be used with the Dynamic Query algorithm presented in the previous Section by individually considering all the terms identified with a confidence s greater than threshold t.

5.3 Evaluation and Discussion

We have evaluated document retrieval and passage retrieval in the context of its usability for QA. For a proper evaluation of SDR we use a corpus of spontaneous speech documents with both manual and automatic transcripts. Manual transcript is an upper bound of the system performance and allows to calculate the drop off due to word error rate.

5.3.1 Experiments

We have conducted experiments using a corpus of 224 transcripts (more than 50,000 words) of automatically transcripted speeches from the European Parliament and the Spanish Parliament. Automatic transcripts have an average word error rate of 26.6%.[5] For each language (English and Spanish), 76 factoid questions were created and 76 keyword sets (i.e. pairs of word/salience) were extracted from them using the procedure described in Section 4.2.

These keyword sets have been used to test several SDR techniques with the transcriptions. We expect that the correct answer to the original question is contained in one or more of the documents returned in the Top-n by the IR engine. In this setting we are not judging the relevance of the documents to a certain topic but the number of queries returning the answer over the total number of queries. We have experimented with document retrieval and passage retrieval.

[5] Transcripts where provided by TALP Research Centre within the framework of TC-STAR project. These transcripts are not the same ones used in the T1 scenario of QAst 2009 (Section 2.3.4).

	Okapi BM25			Vector Space Model			Divergence from Rand.		
System	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
DQ$_{ref}$	**84.21**								
DQ$_{auto}$	**57.89**								
WORD$_{ref}$	43.92	57.25	65.10	36.86	52.15	60.39	45.88	59.60	**67.45**
WORD$_{auto}$	38.03	51.37	54.50	31.37	49.02	54.90	36.46	52.94	**56.07**
3GCH$_{auto}$	16.47	52.94	**65.10**	8.84	34.50	50.19	10.98	46.67	59.29
3GPH$_{auto}$	23.53	47.45	**58.82**	8.62	30.58	44.31	13.72	41.96	56.07
PHAST$_{auto}$	48.62	71.37	**75.29**	31.37	56.47	65.47	46.67	67.06	72.15

Table 5.3: Results in percentage of the document retrieval success for each ranking scheme and term detection method

5.3.2 Evaluation of Document Retrieval

We call DQ$_{ref}$ to the baseline ranking algorithm over reference corpus, while DQ$_{auto}$ is the same algorithm over the automatic transcribed corpus. The difference between both shows the performance fall-out due to ASR action.

We have compared four different methods for term detection that are used in the literature. These methods are used to search the keywords in the documents before ranking them:

- Words (WORD): identifies terms using strict orthographic identity.
- 3-grams of characters (3GCH): the terms are split in overlapping 3-grams of characters.
- 3-grams of phones (3GPH): the terms are transcript to phones and searched as overlapping 3-grams of phones.
- PHAST: uses our new approach described in Section 5.2.2.

These systems for term detection have been used in combination with DQ and three other document ranking functions: Okapi BM25 (BM25 [Robertson et al., 1995]), vector space models (VSM [Salton and Buckley, 1987]), and divergence from randomness (DFR [Amati and Rijsbergen, 2002]).

We have conducted a 5-fold cross-validation. For each fold the full question was randomly split into two subsets: a development set of 25 questions and a test set of 51 questions. For each fold the best parameter setting was selected and applied to the test set. The best parameters for each ranking function were the following.

- BM25: values in $(0, 1]$ for a and $[0, 0.1]$ for b (see [Robertson et al., 1995]).
- DFR: best model has been $I(n)LH_1/H_2$ in almost any experiment (see [Amati and Rijsbergen, 2002]).

- VSM: the nsn scheme was the best in almost any experiment (see [Salton and Buckley, 1987]).
- PHAST: has two tunable parameters, the relevance threshold t and the substring length n. From an empirical basis, we have fixed PHAST's parameters to $t = 0.80$ and $n = 4$ for both passage and document retrieval experiments.

We have chosen Top-n as evaluation measure. It is a measure of precision defined as the number of queries returning a gold document within the top n results of the ranking. The baseline system does not return a ranked list of documents but an unordered set of documents judged relevant. This is why only one result has been reported in Table 5.3 for DQ. DQ$_{ref}$ returned an average of 3.7 documents per query and DQ$_{auto}$ returned an average of 5.7 documents per query. To achieve appropriate comparisons between all proposed techniques, we have chosen Top3 and Top5 as our main evaluation measures. We also provide Top1 for the sake of completeness. In this setting, precision and recall measures are equivalent since we are interested in how many times the IR engine is able to return a gold document in the top 3 or 5 results.

Table 5.3 shows the results of the holdout validation for the three ranking schemes and five term detection methods. The baseline system DQ has been used with reference manual transcripts (DQ$_{ref}$) and with automatic transcripts (DQ$_{auto}$). Also traditional word-based retrieval has been tested over the reference and automatic transcripts as WORD$_{ref}$ and WORD$_{auto}$ respectively. The n-gram based retrieval has been used over the automatic transcripts (3GCH$_{auto}$ and 3GPH$_{auto}$). PHAST obtains better results than any other system working on automatic transcripts. The results are discussed in terms of Top5 for an easier comparison with DQ. Similar conclusions may be achieved with Top3.

Precision loss between DQ$_{ref}$ and DQ$_{auto}$ is 26.3 points. This is due solely to the effect of the ASR transcription. For WORD$_{ref}$, the best result is 67.45%, 16.5 points behind DQ$_{ref}$. With automatic transcripts WORD$_{auto}$ loses 21.3% with respect to WORD$_{ref}$, this loss is comparable to the 26.3% for DQ. The best result of WORD$_{ref}$ (at Top5) is still worse than DQ$_{ref}$, that supports what stated in the previous Section: better results in QA-oriented retrieval are achieved with DQ rather than traditional ranking techniques.

The family of n-gram systems outperforms WORD$_{auto}$ and DQ$_{auto}$ by almost 10 points, but they are still 2 points behind WORD$_{ref}$ and 19 behind DQ$_{ref}$. In terms of Top1 and Top3, n-gram scores are behind WORD$_{auto}$ ones. PHAST outperforms DQ$_{auto}$ in 18.7 points and it is behind DQ$_{ref}$ by 10.5. In Top3, PHAST has still the best performance overall, 15.5 points behind DQ$_{ref}$. PHAST also outperforms 3GCH$_{auto}$ by 10 points, 3GPH$_{auto}$ by 17 and WORD$_{ref}$ by 7.8.

PHAST is better than WORD, 3GCH and 3GPH approaches in two aspects. When the ASR misrecognises one of the keywords (e.g., a proper name) it is impossible for WORD to find this term, and this information is lost. Thus, PHAST outperforms WORD in term matching capabilities allowing an approximate matching of terms. This implies a raising in coverage. The n-gram approach improves coverage and allows approximate matching as PHAST does, but it has no control over n-grams distribution in the text, so it lacks of a high precision (3GPH and 3GCH only outperforms WORD at Top5). PHAST provides more precise and meaningful term detection.

System	Precision	Recall	Passages
DQ_{ref}	86.56%	76.31%	3.78
DQ_{auto}	46.77%	38.15%	5.71
DQ_{PHAST}	64.61%	55.26%	3.80

Table 5.4: Results of passage retrieval. Precision, recall and average number of passages returned per query

5.3.3 Evaluation of Passage Retrieval

After evaluating document retrieval, we have experimented with passage retrieval in the context of QA. Table 5.4 shows the results of our experiments in passage retrieval for spoken documents. DQ_{ref} and DQ_{auto} are the baseline algorithm over manual reference transcripts and automatic transcripts respectively. DQ_{PHAST} is the same baseline using PHAST algorithm for term detection. "Recall" measures the number of queries with correct answer in the returned passages. "Precision" is the number of queries with correct answer if any passage is returned.

There is a 40 point loss between automatic and manual transcripts in precision and recall. In average, DQ_{ref} has returned 3.78 passages per query while DQ_{auto} has returned 5.71. In automatic transcripts DQ_{auto} obtains worse results even returning more passages than in reference transcripts. This is due to the fact that DQ_{auto} drops more keywords (uses an average of 2.2 per query) to build the passages than DQ_{ref} (uses an average of 2.9). Since a substantial number of content words are ill–transcribed, it is easier to find a passage containing n keywords than containing $n+1$. In fact, DQ_{auto} only uses just one keyword in 24 queries, while DQ_{ref} does it in 10 queries.

These results show how term detection is decisive for passage building. The difference between DQ_{auto} and DQ_{ref} in passage retrieval is 40% while it is "only" 29% in document retrieval. Passage retrieval adds a new constraint to the task of document retrieval: the keywords must be close together to be retrieved. Therefore, any transcript error changing a keyword in the transcript may prevent the formation of a passage. Because of its lack of redundancy, passage retrieval is less robust than document retrieval.

DQ_{PHAST} returns an average of 3.80 passages, almost the same as DQ_{ref}, using 2.69 keywords. It surpasses DQ_{auto} by 18% in precision and 17% in recall, taking an intermediate place between DQ_{auto} and DQ_{ref}. The differences among DQ_{PHAST}, DQ_{auto} and DQ_{ref} are similar in passage and document retrieval.

5.4 Related Work

As previously stated in Section 2.4, to the best of our knowledge, only one work addresses the task of passage retrieval for QA on spoken documents using phonetic information.

In Reyes-Barragán et al. [2009], INAOE uses the popular Soundex algorithm to convert documents and question words into sound codes. These codes are indexed together with the words, and are then used during retrieval as a field of the query, having less weight than the original words. What Soundex does, in essence, is to reduce the word to the initial letter followed by the next three consonants in the word. The consonants are grouped into classes and denoted by a one-digit number. For example, names such as Hovy, Moldovan and Prager are converted into H100, M431 and P626 codes. This procedure is similar to applying a hash function to a string.

The manual transcripts of the EPPS corpus have a total of 34,273 words, of which 4,227 are different. From these, 2,110 occur only once (49.9%). After converting all the words to Soundex codes, there are 1,560 different codes, with 431 occurring only once (27.6%). Thus, using Soundex reduces the retrieval search space to almost one third of the original, and merges half of the unique words with other words. The effect of Soundex is the same as performing a clustering of the words with a measure of phonetic distance (like that used in PHAST), with the difference that this does not capture phonetic similarity but orthographic similarity.

Reyes-Barragán et al. [2009] evaluated the effect of Soundex over the whole process of question answering, and reported that there is no improvement in answer accuracy. We believe that this is because Soundex serves as a measure of orthographic similarity (superseding stemming and lemmatization in most of the words), whereas the ASR errors are phonetically, rather than orthographically, similar to the original words.

6. Named Entity Recognition and Classification

Sibyl's strategy for the task of factoid question answering requires to recognize named entities in the documents collection. The Answer Extractor module selects candidate answers from the set of named entities that occur in the passages retrieved by the Passage Retrieval component. The task of detecting named entities is called Named Entity Recognition and Classification (NERC, in short), and is performed using natural language processing techniques. In this Chapter, we detail the strategies that Sibyl uses for NERC of both manual and automatic transcripts.

6.1 NERC in Sibyl

We have taken a machine learning approach to this problem that works in two steps: an entity recognition model and an entity classification model. First, we apply learning at the word level to identify candidates using a **BIO** tagging scheme. Each word is labelled with one tag: a **B** when it is the Beginning of a new named entity, an **I** when it is Inside one, and **O** when it is Outside any named entity. In the second step, the detected named entities (i.e., tag sequences of the form **B**-**I***) are classified into specific categories. The named entity categories used in our QAst experiments are: 'date', 'location', 'measure', 'number', 'organisation', 'person' and 'time'. Each function is modelled with averaged multi-class Perceptron [Crammer and Singer, 2003]. We choose this two step approach instead of a combined model (e.g. having a **BIO** tagger with several different **B**s, one for each entity type) because it allows us to evaluate the effect of word error rate (WER) in both recognition and classification. Also it is customary in NERC.

As learning data, we have manually labelled the named entities that occur in the QAst corpora. This process was performed as follows. First, human assessors tagged the named entities from the manual transcript. Second, the automatic transcripts were aligned with the manual ones using edit distance. Finally, the entity labels were brought to the aligned words in each automatic transcript.

Since we have a single corpus of a limited size, we have performed cross-validation to train and evaluate/adjust the NERC module. The QAst corpus was randomly split into 5 folds, and a NERC model (both recognition and classification) has been learnt for all subsets of 4 folds, with and the remaining fold being labelled using this model. Thus, we can train our NERC with documents from the same domain and kind of orality, but evaluate its performance on new and unseen data.

The task of named entity recognition on speech presents additional difficulties to the written-text task (e.g., OOV words, false positives of named entities…). Many researchers have considered this problem by following either the classical text-based approach to NERC, several ASR-dependent techniques, or even integrating it inside of the ASR system. One interesting way of overcoming the transcription errors consists of not using the usual 1-best transcript (the most probable word sequence, as defined in Equation 5.1 from Section 5.2), but taking the n-best transcripts in order to maximise the detection recall. A direct extension of this approach is to use the ASR internal lattice of all possible transcripts. A joint ASR--NERC model can be applied to the graph, determining both where the named entities are and how they are transcripted, as shown in Favre et al. [2005]. Classical text-based approaches to NERC usually enrich their recognition models with speech related information (e.g., representing named entities as syllables instead of words [Paaß et al., 2009]) or adapting rule-based NERCs to the new corpora [Brun and Ehrmann, 2009].

Since the QAst corpora do not provide n-best lists or full lattices, we have taken a classical approach to the task [Tjong Kim Sang and De Meulder, 2003]. Our NERC system uses a rich set of lexical and syntactic features which are standard in state-of-the-art NERC systems. These features include: words, lemmas, part-of-speech tags, word affixes (i.e., suffixes and prefixes), flags regarding the presence of numerals and capitalisation, use of gazetteers, and 3-grams of these features in a 3 word window. Features with a frequency smaller than 5 are filtered-out to avoid sparsity. Features especially designed for automatic transcripts have been added to the sets *asrA*, *asrB*, and *asrC*. These features are based on phonetic transcription of the words instead of orthographic information. For each word, the following features are added:

- Affixes of phones.
- Phonetic similarity with words in the gazetteer.
- A clustering of the phonetic transcriptions of the words, which was done by grouping words with similar pronunciation. This clustering reduces the sparseness of the word-based features by mapping the words into several smaller subsets of different granularity.
- A clustering of the phones affixes and infixes of each word. This is used to map the words into a sequence of three clusters, having a similar effect as the previous feature.

  Considering the possibility of splitting and merging adjacent words can help to compensate the effect of the usual ASR recognition errors (c.f. Section 5.2). This is achieved by forming new sequences of affix clusters joining current ones with others from previous and following words. These sequences are also added as features for the current word.

6.2 Evaluation and Discussion

Tables 6.1—6.4 show the results of our NERC on the four QAst collections. The results are broken-down by entity types with average measures on the last row. We report the standard NERC measures: precision, recall and the $F_{\beta=1}$ harmonic mean of both. When adding the previously described phonetic features for the ASR transcripts, the overall $F_{\beta=1}$ score improves significantly but no more than 2 points in data-sets *asrA*, *asrB* and *asrC* (Paaß et al. [2009] reports an improvement of 1% for German broadcast news ASR transcripts when using syllables).

We are not aware of other NERC experiments with the EPPS data that we can compare our results with. An $F_{\beta=1}$ value of above 85 is expected for NERC on newspaper texts [Tjong Kim Sang and De Meulder, 2003], but results of 70 have been reported for manual transcripts of the Switchboard corpus of spontaneous speech [Surdeanu et al., 2005]. Our NERC results on manual transcripts are very similar to those obtained with the Switchboard corpus, even though the EPPS corpus is much smaller (900,000 words versus 35,000 words). There are 2,141 named entity examples in the manual transcripts. Of these, 1,374 are either *person* or *organisation* names. The results for these two types

Entity	Prec.	Recall	$F_{\beta=1}$
date	63.95%	63.95%	63.95
location	63.12%	62.17%	62.64
measure	33.33%	28.57%	30.77
number	54.17%	47.27%	50.49
org.	73.51%	78.54%	75.94
person	80.72%	74.41%	77.44
time	57.64%	43.98%	49.89
Overall	70.63%	68.19%	69.39

Table 6.1: NERC on Manual transcripts

Entity	SER	Prec.	Recall	$F_{\beta=1}$
date	30.23%	58.44%	52.94%	55.56
location	16.48%	57.96%	49.43%	53.36
measure	23.81%	16.67%	15.79%	16.22
number	16.36%	50.00%	33.33%	40.00
org.	9.01%	67.21%	67.21%	67.21
person	43.94%	67.21%	56.34%	61.29
time	11.65%	55.41%	33.33%	41.63
Overall	20.04%	63.57%	55.26%	59.13

Table 6.2: *asrA* transcripts, WER=10.6%

Entity	SER	Prec.	Recall	$F_{\beta=1}$
date	40.70%	60.26%	54.65%	57.32
location	24.34%	57.27%	49.24%	52.95
measure	42.85%	18.18%	10.00%	12.90
number	23.63%	52.17%	33.03%	40.45
org.	23.11%	65.53%	64.10%	64.81
person	43.16%	63.03%	51.65%	56.78
time	19.92%	47.59%	27.38%	34.76
Overall	28.58%	61.51%	52.28%	56.52

Table 6.3: *asrB* transcripts, WER=14.0%

Entity	SER	Prec.	Recall	$F_{\beta=1}$
date	46.51%	52.50%	47.19%	49.70
location	30.71%	56.89%	35.98%	44.08
measure	33.33%	5.26%	5.26%	5.26
number	22.72%	50.00%	31.19%	38.42
org.	28.10%	66.11%	55.54%	60.37
person	66.21%	52.47%	39.38%	44.99
time	31.20%	53.33%	28.70%	37.32
Overall	38.72%	58.62%	43.92%	50.22

Table 6.4: *asrC* transcripts, WER=24.1%

are generally better than for the other types, showing that the EPPS corpus is too small for our machine learning approach. Better results can be expected with larger data-sets.

Tables 6.2—6.4 show that, as the transcript WER increases, the scores consequently drop. The main issue with automatic transcripts is the loss of recall. We can see that roughly 1% of recall is lost for every 1% of WER. On *asrC*, the recall falls to 43%, thus it detects 25% fewer named entities than on manual transcripts. A high precision system, such as that of Brun and Ehrmann [2009], reports a 29% decrease in $F_{\beta=1}$ between manual and automatic transcripts (WER=12.11%) of French broadcast news.

In addition to WER, we introduce the analogous measure of Slot Error Rate (SER) defined by Makhoul et al. [1999], which seems more appropriate for evaluating NERC with respect to the ASR performance. For the particular problem of NERC in speech transcripts, the concept of "slot" means each named entity found in a reference manual transcript. A slot is considered correct if it is correctly transcribed by an ASR. Thus, SER measures the rate of named-entity transcription errors by an ASR.

Overall, SER and WER scores behave similarly across the three ASRs (the former doubles the latter), but an analysis according to entity categories shows some differences. We note that some categories are easier to transcribe (i.e. location, number, organization, time), whereas others have consistently above-average SERs (i.e. person and date). This also demonstrates that the ASRs have different recognition capabilities: *asrC* can recognize numbers and measures better than *asrB*, which has the overall best results for the person type. The values of SER and $F_{\beta=1}$ for each entity category are clearly related, although SER cannot account for a total explanation of the results. Named-entity detection relies heavily on the context words surrounding the entity, not only the current entity

Transcript	WER	SER	Proposed	Correct	Precision	Recall	$F_{\beta=1}$
M	—	—	2067	1622	78.47%	75.76%	77.09
asrA	10.6%	20.04%	1850	1319	71.30%	61.98%	66.31
asrB	14%	28.58%	1811	1247	68.86%	58.52%	63.27
asrC	24.1%	38.72%	1566	1013	64.69%	48.47%	55.42

Table 6.5: Results of the Named Entity Recognition without Classification for each transcript

Gold→ Predicted ↓	Precision	Recall	$F_{\beta=1}$	date	loc	mea	num	org	per	time
date	88.71%	90.16%	89.43	55	0	0	4	0	2	0
location	76.85%	74.77%	75.80	1	166	0	0	43	6	0
measure	75.00%	54.55%	63.16	0	0	6	0	0	0	2
number	94.55%	92.86%	93.69	2	0	1	52	0	0	0
organisation	89.72%	93.29%	91.47	3	51	0	0	681	24	0
person	97.24%	92.16%	94.55	0	5	0	0	6	388	0
time	95.12%	98.32%	96.69	1	0	4	0	0	1	117
Overall	90.32%	90.32%	90.32	61	222	11	56	730	415	119

Table 6.6: Confusion matrix of the Named Entity Classifier for Manual reference transcripts

text, and some categories have too few examples to infer reliable conclusions (e.g. 21 measure, 86 date, and 110 number entities).

Table 6.5 evaluates the recognition step of our NERC module alone. An entity is considered to be correct only if its boundaries completely overlap with those of the reference. The numbers confirm that our system has a very poor detection recall and that its results correlate with WER. Table 6.6 is a confusion matrix of the named entity classifier when used on the manual reference transcripts. Remarkably, it has an $F_{\beta=1}$ above 90. Confusion matrices for the other collections (not shown) are very similar to this, presenting very stable $F_{\beta=1}$ scores with respect to WER: 89.84 (*asrA*), 89.90 (*asrB*), 91.12 (*asrC*).

The influence of the ASR on the entity recognition model is greater than on the entity classification model. The study of confusion matrices does not reveal any special source of errors for the classification model. This means that classifying the entities is *easy* once they are detected, probably because only the *easy* ones are detected.

7. Answer Extraction

The Answer Extractor module identifies the exact answer to the given question within the retrieved passages. This process usually involves pattern-matching between information extracted from the question and information extracted from the passage. In Sibyl, answer candidates are identified as the set of named entities occurring in the retrieved passages that have the same type as the answer type detected by the Question Processing module (Chapter 4). These candidates are then ranked using several different measures: the two main scoring functions are a set of heuristics that measure keyword distance and density, and a ML reranker that matches the syntactic structures between question and answers. The output consists of a list of the highest ranked answers.

In the rest of this Chapter we describe the Answer Extraction module. This module uses some tools designed for written text and machine learning techniques that substantially improve the answer extraction quality. First of all, we describe a baseline system (7.1) for the extraction that uses only shallow information and the manually developed heuristic combination. This baseline is then improved (7.2) with a reranker based on machine learning. Later, we use a dependency parser to get more informed features for the ranking (7.3). As we will see, this improves the precision and coverage of the extraction (7.4). Finally, we present and discus several answer extraction methods form the literature that are related to our work (7.5).

The main content of this Chapter has been previously published in the conference paper [Comas et al., 2010].

7.1 Heuristic Answer Extractor

We will refer to the baseline version of the Answer Extractor as Heuristic Baseline. This is based on the properties of the context in which the candidate answers appear in the retrieved passages. The candidates are ranked using a scoring function based on a set of seven heuristics that measure keyword distance and density. The heuristics are inspired by those of Surdeanu et al. [2006]:

H_1 *Same word sequence*: computes the number of words that are recognised in the same order in the answer context.

H_2 *Punctuation flag*: 1 when the candidate answer is followed by a punctuation mark, 0 otherwise.

H_3 *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.

H_4 *Same sentence*: the number of question words in the same sentence as the candidate answer.

H_5 *Matched keywords*: the number of question words found in the answer context.

H_6 *Answer span*: the largest distance (in words) between two question keywords in the given context.

H_7 *Distance from QFW*: measures the distance between the candidate answer and the question focus word (QFW). This is only enabled for certain question types.

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

$$\text{score} = H_1 + H_2 + 2H_3 + H_4 + H_5 - \frac{1}{4}\sqrt{H_6} - H_7, \tag{7.1}$$

where the heuristic weights are manually optimised following Surdeanu et al. [2006]. This score is used to rank the candidate answers. The top ranked candidates are selected as the final answers to the question (the best 5 in QAst experiments).

7.2 Heuristic Reranker

The previously described Heuristic Baseline is dependent on the document collection, because the weights must be adjusted according to the characteristics of the documents. This is not an easy task since it involves the optimisation of several variables at the same time. In addition, this is a rough score that does not make any use of simple information such as the expected answer type or the repetition of candidates. Thus, we have implemented a machine learning layer that takes the seven heuristic scores and a set of development questions, and learns how to combine them to rerank the candidates. We call this approach Heuristic Reranker, and it makes it unnecessary to manually tune the weight of the heuristics as the combinatorial function is acquired by the learning algorithm from the development examples. Additionally, this process allows us to include arbitrary new information in the extraction process coded as features.

We have used binary Support Vector Machines (SVM) for learning a classifier to distinguish correct from incorrect answer candidates.[1] Each candidate answer obtained from the development questions using the Heuristic Baseline is now a training example. The candidates may be either right or wrong answers, being either a positive or negative example in the SVM model. For each candidate, the following set of features is computed:

1. Score value (Equation 7.1)
2. Order in the ranking according to the score value
3. Values of each heuristic $H_1 \ldots H_7$
4. Number of times this candidate is repeated
5. Length in number of words
6. Candidate's named entity type
7. Number of keyword in the query.

These features are converted into binary features before the learning process. To do this, we proceed as follows: first, the range of values of each feature is split into k parts, numbered from 0 to $k-1$. Each value is then expanded with a series of inequalities framing it. For example, if the candidate answer is 3 words long (feature *CAlen:3*), these new features are added: *CAlen>0*, *CAlen>1*, *CAlen>2*, *CAlen<4*...*CAlen<k-2*, *CAlen<k-1*. Categorical variables (i.e., type of named entity) are not binarised.

The learnt model is used to rerank the output of the Heuristic extractor. The candidates are sorted according to the SVM score they have obtained, and this is the final output of the Answer Extractor.

Note that we have not used a ranking-SVM but a standard binary classifier; we are not concerned with having multiple correct answers in the top positions or the quality of the overall ranking (like in an IR scenario), but with having the first correct answer in the topmost position. Our reranking problem can be better modeled as a classification problem; as there are only two possible levels (for “correct” and “incorrect” answers), the reranking problem is a pure classification problem, and we use a binary classifier for simplicity as the ranker learns a more complex classifier.

[1] http://svmlight.joachims.org

A ranking-SVM [Joachims, 2002] learns a model trained on all possible pairs between examples in the example rankings. It proceeds taking ordered pairs (e.g., $(e1, e2)$ with $r_{e1} < r_{e2}$) as positive examples for learning. Due to example representation, what it optimises is the difference of labels ($l_{e1} - l_{e2}$) according to the difference of feature vectors ($w_{e1} - w_{e2}$). Even using only two labels, the representation of the problem is more complex than what is needed in this task.

7.3 Adding Syntactic Information

Keyword density measures from Section 7.1 do not capture meaning. Arguably, a successful QA system should use syntactic and semantic information to understand the text and make deductions from it, but this is even more difficult in speech transcripts. As a key contribution we add syntactic information to our answer extractor to improve its ability to distinguish correct and incorrect candidates. Syntax allows us to model the relations between words, which is more powerful than simple density measures. We call this approach Syntactic Reranker.

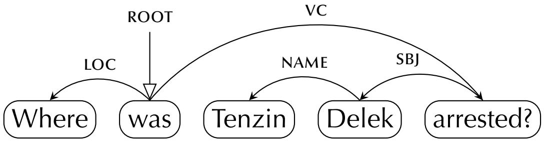
We have labelled the collection with syntactic relations using an in-house dependency parser [Lluís et al., 2009];[2] thus, any pair of words in a sentence can be linked by a sequence or path of syntactic relations extracted from the dependency tree. We have also labelled the questions with syntactic relations. Our dependency parser has been trained with the CoNLL—2007 Shared Task collection, a collection of newspaper texts [Nivre et al., 2007]. Adapting the parser to speech is beyond the scope of our work, we focus only on extracting robust features to make the parser useful for speech.

The key assumption is that the syntactic relations between the keywords and the candidate answer (in the collection) should be similar to the syntactic relations between the keywords and the question tag in the question. This denotes that keywords in the text are framing the candidate answer with restrictions similar to those expressed in the question. For factoid questions, this question tag is either *who*, *where*, *when*, *how*, *what*, or *which*. Comparing the paths should help in disregarding candidate answers that are near the keywords but not properly related to them, and to get candidates that are long-distance syntactic relations (i.e., those that cannot be captured with local heuristics).

For example, consider the question "*Where was Tenzin Delek arrested?*," Figure 7.1 shows the path of syntactic relations joining the question tag *Where* with the the keywords *Tenzin* and *Delek*. Figure 7.2 shows the parsing of two sentences from the EPPS collection that contain the candidate answers *Scotland* and *Tibetan China*. Our heuristic measures based on keyword density cannot distinguish between the correct one (*Tibetan China*) and the totally unrelated one (*Scotland*). In Sentence 1, *Scotland* is a locative nominal modifier of *constituents* but is not related to *Tenzin Delek*. Prior to comparison, the paths are simplified to avoid sparsity removing frequent labels that are of little use like name modifiers (NMOD), preposition modifiers (PMOD), and proper name modifiers (NAME).

[2] `http://www.lsi.upc.edu/~xlluis/jointparser/`

Question9: Where was Tenzin Delek arrested?



Path from “Where” to “Tenzin_Delek:” LOC - ROOT - VC - SBJ - NAME
Simplified path: LOC - VC - SBJ - NAME

Figure 7.1: Sample question with dependency parsing

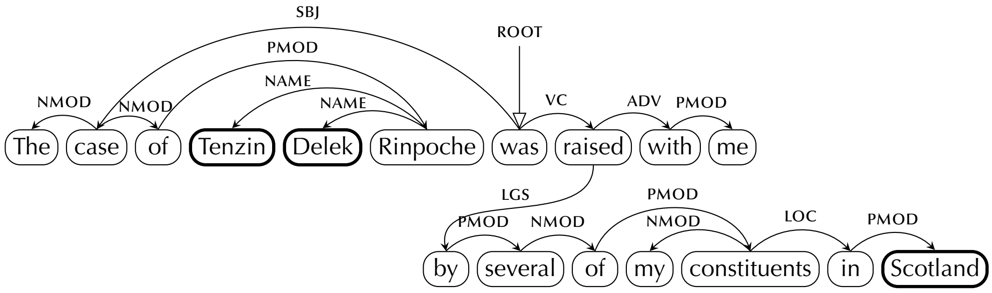
Sequences of contiguous verbs are represented by a single VC label. Note that the label ROOT denotes the main verb of the sentence; we transform this into VC because there is no need to force the paths to contain the *main* verb of the sentence. If we compare the path from the question with the paths from Sentence 1 (shown in Figure 7.2), we can see that the latter differs in its extra LGS relation. LGS denotes the logical subject of a passive verb; this means that *Scotland* modifies a noun phrase that has a syntactic relation with the main verb other than a locative modifier. Thus, *Scotland* is not necessary expressing a locative restriction of a verb whose subject is *Tenzin Delek*. Instead, if we look at Sentence 2, we find one extra LOC relation, which means that *Tibetan China* is a locative modifier of a locative modifier whose subject is the keyword *Tenzin Delek*.

To compare two given paths Q_k and T_k, where Q_k has been extracted from the question for keyword k and T_k from the collection, we use a dynamic programming algorithm to align them. It finds the longest sequence (M_k) of labels that can be matched without changing their order, and then computes the labels from Q_k that are missing in T_k and vice-versa. This information is summarised in a set of features that enrich the model described in the previous section. We aim to provide very robust features, since the result of parsing speech transcripts can be very poor. For each candidate answer, the features from these 5 classes are added:

1. Number of syntactically related keywords and their proportion with respect to the total number of keywords.
2. Distance from candidate answer to keywords in number of syntactic relations traversed in the path between them.
3. The length of M_k and the ratio $|M_k|/|Q_k|$ for each keyword k. Also, the maximum, minimum and average of these measures in all keywords.
4. Total number of matched labels: $\sum_{\forall k} |M_k|$.
5. Total number of labels inserted and the count for each different label.

Before learning the SVM model, the ratios are discretised to intervals and integer values are expanded to binary features as explained in the previous section. As with the

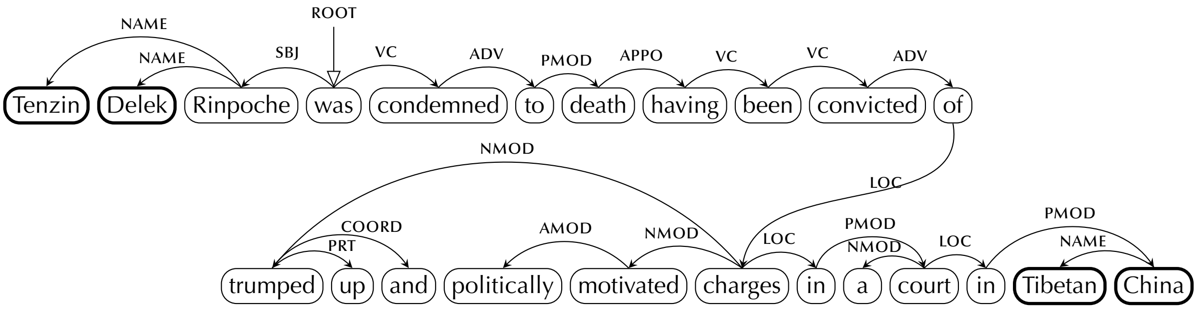
Sentence number 1



Path from *Scotland* to *Tenzin Delek*: PMOD - LOC - PMOD - NMOD - PMOD - LGS - VC - ROOT - SBJ - NMOD - PMOD - NAME

Simplified path: LOC - LGS - VC - SBJ - NAME

Sentence number 2



Path from *Tibetan China* to *Tenzin Delek*: NAME - PMOD - LOC - PMOD - LOC - PMOD - ADV - VC - VC - APPO - PMOD - ADV - VC - ROOT - SBJ - NAME

Simplified path: LOC - LOC - ADV - VC - ADV - VC - SBJ

Figure 7.2: Two examples of candidate answers and dependency parsing

Heuristic Reranker, the learnt model is used to rerank the baseline answers into a final ranking.

Continuing with the previous example, consider the sentence number 2 in Figure 7.2. To obtain the features of candidate answer *Tibetan China*, first we search for the question keywords in the sentence (only one is found, *Tenzin Delek*), and then the simplified paths that go from keyword to candidate answer are aligned this way using the edit-distance algorithm:

$$\begin{array}{rccccccc} Q_1: & - & \mathrm{LOC} & - & \mathrm{VC} & - & - & \mathrm{SBJ} \\ T_1: & \mathrm{LOC} & \mathrm{LOC} & \mathrm{ADV} & \mathrm{VC} & \mathrm{ADV} & \mathrm{VC} & \mathrm{SBJ}, \end{array}$$

thus, it yields the following features for candidate *Tibetan China* in sentence number 2:

Related keywords: $\mathit{KWfound} = 1$, $\mathit{KWratio} = 0.5$.

Relations path length: $\mathit{PathLength}_1 = 14$ (using the full path).

Length of common labels: $M_1 = 3$, $|M_1|/|Q_1| = 1$.

Matched Labels: $\sum_{\forall k} |M_k| = 3$.

Inserted labels: $\mathit{Insert}_{ADV} = 2$, $\mathit{Insert}_{LOC} = 1$, $\mathit{Insert}_{VC} = 1$, $\mathit{Insert}_{all} = 4$.

Since that there is only one keyword in this example, the maximum, minimum and average values of the $|M_k|$ measures are all the same and are not shown here.

7.4 Evaluation and Discussion

The word presented in Chapters 4 through 7 describes a fully functional QA system that can be evaluated as a whole. In this section we have evaluated the performance of Sibyl in the context of the QAst evaluation [Turmo et al., 2009], using the TEST set questions with manual transcripts of the European Parliament Plenary Sessions (EPPS) corpus. Section 7.4.1 describes the experiments and Section 7.4.2 describes the measures used. In Section 7.4.4, the results of the reranking methods from this Chapter are discussed and compared with the results achieved by the participants in the 2009 QAst evaluation (Section 2.3.4.3).

7.4.1 Experiments

In these experiments we have used the English data-sets provided by the QAst evaluation campaign from year 2009, since it is the most recent and complete of the QAst evaluations (Section 2.3.4). This data is a collection of manual transcripts of 3 hours from the European

Parliament Plenary Sessions (EPPS) in English. This is about 35,000 words long. In the manual transcripts, the sessions are divided in turns according to the speaker. The only punctuation mark is the full stop. There are also marks for hesitations and partial words and most of the names are capitalised. The ASR transcripts *asrA*, *asrB* and *asrC* do not have any kind of capitalisation, punctuation or turn division. These automatic transcripts come from the TC-STAR evaluation and were selected to provide the widest possible range of WERs.

For the experiment with the Syntactic Reranker, we require to have the dependency parsing of documents and question. For the dependency parser it is utterly important to have sentence boundaries in the text, but these boundaries do not exist on automatic transcripts. To segment the text in sentences, we have aligned manual transcripts with automatic transcripts using the edit distance and transferred the full stops from the former to the latter. These enhanced automatic transcripts are used only in the Syntactic Reranker experiments. In a real application, without available manual transcripts, this alignment is infeasible and the method would require an automatic sentence splitter. Thus, the results of the Syntactic Reranker for automatic transcripts showed in this Section must be considered an upper bound of what could be achieved in a real application. It has been shown by Paulik et al. [2008] that it is feasible to build an automatic sentence splitter for the EPPS corpora which can be useful for other tasks.

We have used the question set A (see Section 2.3). There are two subsets of questions: a development set (DEV) of 50 questions and a test set (TEST) of 100 questions. The DEV questions have been used to adjust some parameters and to learn the reranking models. Each subset contains factoid and definitional questions in a proportion of 75%–25% respectively. Some of the questions do not have an answer in the collection, therefore the correct answer for them is "nil." Given that SIBYL is a factoid QA system, we have only experimented with the 75 factoid questions from TEST set, from those 11 are "nil" and 64 have a concrete answer. All results in this section are referred only to factoid questions.

We have decided not to use EPPS questions from the 2008 QAst evaluation in the DEV set. It has been reported by Bernard et al. [2010] that in the 2009 evaluation, the correct answers are found further from the keywords than in the 2008 evaluation. The 2009 questions are more difficult to tackle for our approach, and we believe that the models learnt with 2008 data would be of little use in 2009 questions.

7.4.2 Measures

Our evaluation reports the same measures as the official QAst evaluation. The QA system outputs a ranking of at most 5 answers for each question. If an answer is incomplete or it includes more information than necessary, or the document does not provide the justification for the answer, the answer is considered incorrect. Correct answers are evaluated by two measures as previously defined:

- *Mean Reciprocal Rank (MRR):* Average of the inverses of the ranking of the first correct answer for each question. It is defined as $\frac{1}{N}\sum_{i=0}^{N}\frac{1}{rank_i}$, where $rank_i$ is the position of the first correct answer in the answer list for question number i.

- *Accuracy:* The fraction of correct answers ranked in first position in the list of 5 possible answers.
- *Top1:* The number of questions with a correct answer in the first position in the list.
- *Top5:* The number of questions with a correct answer anywhere in the 5 answers list.

Questions without an answer in the text are evaluated in the same way as the other questions: the string “nil” is the correct answer, and it may be anywhere in the ranking. Then MRR and accuracy are calculated as defined above.

We report the results only for factoid questions, and we further separated the questions in two groups according to if they have a defined answers or expect a “nil” answers.

7.4.3 Heuristic Baseline Evaluation

Table 7.1 summarises the overall results of our Heuristic Baseline for factoid questions on manual and automatic (*asrA*, *asrB*, *asrC*) transcripts. It shows Accuracy, MRR, Top1, and Top5 scores for each track and run as defined previously[3] Sibyl's Heuristic Baseline is compared with the results obtained by the LIMSI, INAOE, and TOKIO systems from the official QAst 2009 evaluation.[4] See Section 2.4 for a description of their characteristics.

For the manual transcripts, Sibyl's baseline results are worse than those of LIMSI by more than 8 points in MRR and accuracy. INAOE also has a much larger MRR score. Top5 shows that our baseline system has a low coverage, 10 answers behind INAOE (25% less) and 6 behind LIMSI.

We can see that moving from manual transcripts to the *asrA* transcripts has very little impact on Sibyl's MRR and accuracy. A subsequent increase in WER of 13.5% on the *asrC* transcripts has no additional impact (results are slightly better due to correctly recognizing one additional “nil” question). Only on the *asrB* transcripts are our results considerably worse than in the other transcripts. Although *asrB*'s WER of 14% is much lower than for *asrC* and almost the same as *asrA*, this transcript achieves the worst MRR and accuracy results for any of the evaluated systems.

Although our initial results for manual transcripts are worse than those of LIMSI and INAOE, the degradation produced by the ASR transcripts is much lower. Sibyl has a better MRR than INAOE for *asrB* and than LIMSI for *asrC*. These results show that Sibyl is much more robust than the other systems when dealing with ASR transcripts. The MRR score of LIMSI drops by 12 points when moving from the manual to *asrC* transcripts, and that of INAOE drops by almost 9. It must be noted that INAOE's numbers are difficult to compare with our results. As stated in [Reyes-Barragán et al., 2009], they enriched the *asrC* transcripts with named entities extracted from the *asrA* and *asrB* transcripts because of

[3] Our results are slightly different from the official ones due to a major rewriting and bug-correction of Sibyl's source code.

[4] Note that in the official QAst results Turmo et al. [2009], the English EPPS task for written questions is named “T1a.”

		64 questions		11 nil questions			
Transcript	**System Name**	**Top1**	**Top5**	**Top1**	**Top5**	**MRR**	**Accuracy**
Manual	Heuristic Baseline	13	29	3	5	0.3002	21.33%
asrA	Heuristic Baseline	12	26	4	5	0.2993	21.33%
asrB	Heuristic Baseline	10	20	3	5	0.2311	17.33%
asrC	Heuristic Baseline	12	24	5	6	0.3022	22.67%

Table 7.1: Results of Answer Extraction rerankers for SIBYL's Heuristic Baseline

the poor results achieved by their NERC with *asrC*. Thus, their Accuracy and MRR are unreliable measures.

Table 7.2 contains a statistical error analysis of our Heuristic Baseline, covering the Question Processing, Passage Retrieval, and Answer Extraction parts. The analysis only deals with questions with non-nil answers, i.e., questions that have a correct answer in the documents. The meaning of each column is as follows: "#Q" stands for the number of factoid questions, "QC" is the number of questions whose the expected answer type was correctly detected by the Question Processing module, "PR" is the number of questions where at least one passage with the correct answer was retrieved, "QC&PR" counts the number of questions with correct answer type and correct answer retrieved, and "C.NE" is the number of questions where the retrieved passages contain the correct answer tagged as a named entity of the right type (specified by the Question Processing module), so it becomes a candidate answer for the Answer Extraction module. "Top5 non-nil" stands for the number of questions with non-nil answer correctly answered by our system in the Top5 candidates, and the same for "Top1." Finally, the "Avg. Loss" row shows the performance loss (averaged across all transcripts) introduced by each module. Losses for QC, PR, and QC&PR are relative to the Q column, while C.NE, Top5, and Top1 are relative to the previous column.

The figures in Table 7.2 show that in 14 of the questions the expected answer type is misrecognised (QC). Thus, SIBYL cannot answer more than 50 question correctly in any of the scenarios, and the Answer Extractor accuracy can be 78% at best. Our keyword selection strategy cannot be evaluated as a standalone task in the Question Processing module, but it is reflected in the Passage Retrieval evaluation. The PR column shows that the success of our Passage Retrieval module decreases as WER increases. As expected, the more transcription errors, the more difficult the retrieval task. On the manual transcripts, 20.3% of the queries do not retrieve the correct answer. This figure increases to 36% for the *asrC* transcripts. Thus, Passage Retrieval introduces more errors than Question Processing for the ASR transcripts. The specific distribution of QC and PR errors means that only 56% of correctly classified questions retrieved the correct answer.

In column C.NE, we can see how many of the correct answers retrieved by our Passage Retrieval module are annotated with the expected answer type. The small difference between the QC&PR and C.NE columns (an average of 8%) shows that most of the answers are correctly tagged by our NERC. As we have seen in Table 6.1, the $F_{\beta=1}$ score of our NERC is below 70%, but this relatively poor performance does not overly affect

Transcript	#Q	QC	PR	QC&PR	C.NE	Top5 non-nil	Top1 non-nil
Manual	64	50	51	41	41	29	13
asrA	64	50	47	39	35	26	12
asrB	64	50	43	32	26	20	10
asrC	64	50	41	31	28	24	12
Avg. Loss		-22%	-29%	-44%	-8%	-25%	-54%

Table 7.2: Error analysis of the QA system components

the final QA results. Therefore, the NERC module we have developed is useful enough for this task on manual and automatic transcripts. It should be noted that this evaluatin may be biased, as there is a possible disagrrement between the humans who tagged the named entities (for learning the NERC model) and the humans who wrote and evaluated the questions (the latter were implicitly tagging named entities when selecting which entity is the correct answer to a question). In contrast to the previous measures, the lowest C.NE value is achieved on the *asrB* transcripts. This may indicate that the *asrB* transcripts, although having just a 3% higher WER than the *asrA* transcripts, are especially prone to misrecognising named entities, and thus the *asrB* transcripts are the least suitable ASR set for factoid QA of the three.

Finally, we can see from the last two columns that one of the main limitations of Sibyl's Heuristic Baseline system is the poor performance of the Answer Extraction module. More than 25% of the correctly retrieved and tagged answers are not correctly extracted, and only one third of them reach the Top1.

7.4.4 Improving Answer Extraction with Rerankers

As we explained in Chapter 2, the Answer Extraction module is the last one in the QA pipeline. The previous modules may introduce errors that make it impossible to extract the correct answer (e.g., errors in passage retrieval, errors in detecting the expected answer type, etc). Due to these errors, it is only possible to extract the correct answer in 46 of the 75 factoid questions (41 plus 5 “nil” questions, as shown in Table 7.2). This means that 46 is the theoretical upper bound of the Heuristic Baseline answer extractor for the manual transcripts. Thus, Answer Extraction is the major bottleneck of the process. In Sections 7.2 and 7.3, we introduced two more answer extractors, namely Heuristic Reranker and Syntactic Reranker, which were designed to improve the Heuristic Baseline. Table 7.3 summarises the results of each answer extractor, the “Upper Bound” row shows the best results achievable with a perfect Answer Extractor.

Our results show that the Heuristic Reranker is able to learn a better ranking of candidates than the Heuristic approach, in terms of MRR and accuracy, in all transcripts except collection *asrC*. The greatest improvement is in terms of accuracy. For manual

		64 questions		11 nil questions			
Transcript	**System Name**	**Top1**	**Top5**	**Top1**	**Top5**	**MRR**	**Accuracy**
Manual	Upper Bound	41	41	3	5	0.6133	58.67%
	Heuristic Baseline	13	29	3	5	0.3002	21.33%
	Heuristic Rerank	16	29	3	5	0.3360	25.33%
	Syntactic Rerank	19	32	3	5	**0.3687**	**29.33%**
asrA	Upper Bound	39	39	4	5	0.5867	57.33%
	Heuristic Baseline	12	26	4	5	0.2993	21.33%
	Heuristic Rerank	16	25	4	5	**0.3171**	**26.67%**
	Syntactic Rerank	14	26	4	5	0.3089	24.00%
asrB	Upper Bound	32	32	3	5	0.4933	46.67%
	Heuristic Baseline	10	20	3	5	0.2311	17.33%
	Heuristic Rerank	12	20	3	5	0.2476	20.00%
	Syntactic Rerank	14	21	3	5	**0.2760**	**22.67%**
asrC	Upper Bound	31	31	5	6	0.4933	48.00%
	Heuristic Baseline	12	24	5	6	**0.3022**	**22.67%**
	Heuristic Rerank	12	21	5	6	0.2760	**22.67%**
	Syntactic Rerank	7	22	5	7	0.2520	16.00%

Table 7.3: Results of Answer Extraction rerankers

transcripts, Top1 increases from 13 to 16 but coverage (Top5) is not improved. This means that no new correct answers are added to any list, but they are now better ordered. The same happens to *asrA* and *asrB*: Top1 increases but Top5 is maintained or decreases. This may be explained by the fact that the Heuristic Reranker does not include truly new information in its features, it just makes better use of the heuristic scores used to calculate the original Heuristic Baseline ranking.

The Syntactic Reranker improves both the MRR and accuracy of the Heuristic Reranker by more than 3 points for manual transcripts. Both Top1 and Top5 are improved indicating that dependency parsing incorporates useful and new information, and new correct answers appear in the top 5. For the automatic transcripts, the Syntactic Reranker has mixed results. For *asrA*, Top1 and Top5 are improved by more than one point of MRR. For *asrB*, syntax makes little difference to the Heuristic Rerank, whereas for *asrC*, this strategy is clearly harmful to both the Top1 and Top5 scores. In this experiment, the automatic transcripts have been enhanced with punctuation marks, as described in Section 7.4.1. If an automatic sentence splitter had been used, the results would probably have been worse. The SVM classifiers used for the extraction have been learnt with a polynomial kernel of $d = 2$ and a trade-off $c = 100$. These parameters where adjusted on a sample of the DEV questions, although the available data was too small to properly fine-tune c.

It is remarkable that even using the very small training set of 50 training questions (43 are defined), which generates a total of only 37 positive examples (usually only one per question) for 1,600 negative examples, is sufficient to get a significant impact with both methods. To study the effect of the training set size, we have conducted further experiments with an expanded training set as shown in Table 7.4. These experiments are

		64 questions		11 nil questions			
Transcript	**System Name**	**Top1**	**Top5**	**Top1**	**Top5**	**MRR**	**Accuracy**
Manual	Heuristic Rerank	23	33	3	5	0.4036	34.67%
	Syntactic Rerank	24	36	3	5	0.4251	36.00%
asrA	Heuristic Rerank	15	27	4	5	0.3211	25.33%
	Syntactic Rerank	17	29	4	5	0.3453	28.00%
asrB	Heuristic Rerank	13	21	3	5	0.2678	21.33%
	Syntactic Rerank	16	21	3	5	0.2916	25.33%
asrC	Heuristic Rerank	14	24	5	6	0.3044	25.33%
	Syntactic Rerank	12	23	5	7	0.2913	22.67%

Table 7.4: Results of leave-one-out experiments

leave-one-out evaluations, mixing the DEV and TEST sets. For each TEST question, a reranker model has been learnt using all of the examples from the DEV set and all of the examples from the TEST set, except those corresponding to the question itself. Therefore, each training set contains examples from 149 questions instead of just 50 as in the previous experiments, yielding an average of 175 positive and 56,000 negative examples per set. This setting does not bias the models, as all the questions are totally independent.

As we can see in Table 7.4, the results are much better when using larger training sets on manual transcripts: both rerankers show a parallel improvement of 6 or more points; Syntactic Reranker Top5 reaches 41 (36 + 5 "nil" questions) out of 46, 89% of the upper bound. These new results are clearly better than any system in the QAst 2009 evaluation (Table 7.6). The Heuristic Baseline MRR for manual transcripts improves by 41.6% in this experiment (22.8% with the original training set), and this improvement is also better than any reported in the literature for comparable approaches based on syntax (Section 7.5). With the ASR transcripts, the benefits are much smaller than for the manual transcripts, exhibiting an improvement of less than 3 points of MRR and accuracy for the Heuristic Reranker. This demonstrates that, despite the good results achieved by the baseline system, ASR transcripts are consistently more difficult for the QA task. For *asrC*, the leave-one-out evaluation adds 3 points to the Heuristic Rerank accuracy, topping the initial Heuristic Baseline values. The results from the Syntactic Rerank confirm that both WER and training set size are important factors that play opposite roles. In *asrB*, the Syntactic Rerank only improves the Heuristic Reranker when using the expanded training set, whereas for *asrA*, the Syntactic Rerank is better than the Heuristic with both training sets. The Syntactic Reranker results for *asrC* the results are consistently worse than the Heuristic Rerank. Although we do not have a direct evaluation of the dependency parsing with these transcripts, it is clear that high WERs are disruptive enough to render the parsing useless.

As a final remark, it has been suggested by Sun et al. [2005] that answer extraction based on dependency relation matching does not perform well on short questions with few words. Longer questions tend to have more keyword terms and longer dependency relation paths that may be more informative for our ML reranker, thus yielding better

Question Length	#Q	MRR
4–7	35	0.297
8–11	26	0.430
12+	5	0.466

# of KW	#Q	MRR
1	2	0
2	13	0.23
3	25	0.43
4	16	0.31
5	5	0.47
6	5	0.20
7	0	0
8	1	1
9	1	1

Table 7.5: MRR of the Syntactic Rerank on manual transcripts grouped by question length (left) and by number of keywords (right)

results. We have evaluated this issue with Sibyl's Syntactic Reranker for manual transcripts. In Table 7.5 (left) we show the MRR score for non-nil factoid questions grouped in three blocks according question length. The reported bias is observed in Sibyl, as the score is worse for short questions and improves with longer ones, although the numbers are again too small to assess a definitive conclusion. In Table 7.5 (right), the same analysis is performed according to the number of keywords that has been identified in the question. We can se that there is a *sweet spot* at 3 keywords, in contrast with the results of [Sun et al., 2005]. Although very good results are obtained with 5 or more keywords, these numbers account for only the 12% of the questions and its significance is unclear.

7.4.5 Comparison with QAst 2009 Results

Table 7.6 gathers the best runs from the QAst 2009 participants with the results of our rerankers.

The best QA system for manual transcripts (limsi, Table 7.6) has better results than any of our three runs in MRR,[5] but our Syntactic Reranker achieves the same accuracy. This is an important result, because our approach is free of handcrafted or language-dependent rules, and the cost of providing training examples (i.e. question/answer pairs) is low compared to other kinds of annotation, while achieving the same or better accuracy than those that require effort to be manually adapted.

On the automatic transcripts, our results are competitive with the best systems. It is also remarkable that the results for *asrB* are worse than for *asrC* for all of the QA systems. This fact supports our hypothesis from Section 2.3.4 that the transcription error rate is not as important as the distribution of errors. Transcript *asrC* has a higher WER, but is more suitable for factoid QA than *asrB*. In the particular case of Sibyl, we can see

[5] It is not possible to know if this difference is due solely to Answer Extraction, or because the Question Processing and Passage Retrieval modules are better than ours. This would require a white-box evaluation of QAst systems, which is beyond the scope of this paper.

		64 non-nil Q		11 nil Q			
Transcript	**System Name**	**Top1**	**Top5**	**Top1**	**Top5**	**MRR**	**Accuracy**
Manual	Sibyl Baseline	13	29	3	5	0.3002	21.33%
	inaoe	18	39	0	5	0.3824	24.00%
	limsi	20	35	2	7	0.3931	29.33%
	tokio	5	11	0	0	0.1067	6.67%
	Heuristic Rerank	16	29	3	5	0.3360	25.33%
	Syntactic Rerank	19	32	3	5	0.3687	29.33%
asrA 10.6% WER	Sibyl Baseline	12	26	4	5	0.2993	21.33%
	inaoe	15	30	3	5	0.3236	24.00%
	limsi	17	27	4	5	0.3353	28.00%
	tokio	3	14	0	0	0.0849	4.00%
	Heuristic Rerank	16	25	4	5	0.3171	26.67%
	Syntactic Rerank	14	26	4	5	0.3089	24.00%
asrB 14.0% WER	Sibyl Baseline	10	20	3	5	0.2311	17.33%
	inaoe	8	16	4	5	0.2167	16.00%
	limsi	13	20	4	4	0.2660	22.67%
	tokio	3	9	0	0	0.0642	4.00%
	Heuristic Rerank	12	20	3	5	0.2476	20.00%
	Syntactic Rerank	14	21	3	5	0.2760	22.67%
asrC 24.1% WER	Sibyl Baseline	12	24	5	6	0.3022	22.67%
	inaoe	17	23	3	6	0.3076	26.67%
	limsi	14	18	4	5	0.2616	24.00%
	tokio	2	15	2	2	0.1164	5.33%
	Heuristic Rerank	12	21	5	6	0.2760	22.67%
	Syntactic Rerank	7	22	5	7	0.2520	16.00%

Table 7.6: Comparison of the QAst 2009 results for English EPPS with Sibyl's rerankers

from Table 7.2 that as the Passage Retrieval performance worsens, the likelihood of not retrieving the correct answer increases, as well as the chances of not finding a passage or suitable candidates and thus answering "nil" to the question. Given that for *asrC* we got more correct "nil" than in the other scenarios, we believe that our system is improving the recall on "nil" questions while having similar or slightly better base results for non-nil questions. This partially explains why the QA performance improves in *asrC* whereas NERC performance slightly worsens.

In order to outperform the limsi and inaoe systems, we need to increment the reranker's amount of training data. Our leave-one-out experiment from Table 7.4 shows that Sibyl can achieve an MRR of 0.425 for manual transcripts and MRRs of 0.345, 0.2916 and 0.304 for the ASR transcripts when the reranker learns on a three times larger corpus of question/answer examples.

7.5 Related Work

Answer Extraction could be based solely on shallow linguistic information (also called *surface* information), e.g. distance and redundancy measures that favour candidate answers statistically related to the question context, as we used in our Heuristic Baseline method in Section 7.1. But this *bag-of-words* approach has obvious limitations, such as the lack of semantic information and the potential lexical mismatch between question and answer, as defined by Berger et al. [2000].

To overcome these problems, the researchers tried to find representations where the distance between the question and the sentences containing correct answers was small, and where the distance to incorrect answers was large [Echihabi and Marcu, 2003]. These representations involve a more in-depth approach to Answer Extraction, making use of syntactic or semantic distance. The works discussed in the rest of this section use some kind of similarity measure between linguistic structures when comparing questions and potential answers.

Remarkably deep textual processing has been carried by researchers from Language Computer Corp. (LCC). They exploited semantic information, transforming the semantic representations of questions and answers into logic forms, then used a full reasoning engine to infer causality between text and answers [Moldovan et al., 2007b]. This approach needs a huge number of high-quality rules representing real-world knowledge and discourse structure to bring the logic forms together and be useful [Novischi and Moldovan, 2006]. LCC has successfully applied this approach in many different QA tasks over recent years [Harabagiu et al., 2005]. A similar approach is that of Hartrumpf et al. [2009] for the German language. They represent documents as semantic networks by means of a syntactico-semantic parser, and then use semantic calculus to match the question in the nodes of the net.

A simpler, straightforward way of incorporating semantic knowledge was reported by Shen and Lapata [2007]. They used a semantic role labeller to convert question and answer sentences into semantic structures drawn from the FrameNet database. Ambiguity was tackled by converting the structures to graphs and mixing all possible frames. A graph-matching algorithm can then be used to rank the answers according to their semantic similarity to the question. This approach is limited by the small coverage of the FrameNet database, and can be applied to less than 35% of the TREC02–05 factoid questions. Shen and Lapata report that semantic frames boost the precision of the extractor by an average of 50% when added to the dependency relations (for the questions where it can be applied). Another graph-matching strategy with semantic information is that of Bilotti et al. [2010], who employed a semantic role labeller to assign PropBank semantic roles to questions and answers. These roles are then decomposed into atomic constraints, in the form of graphs, using a set of rules. Partial matchings between questions and answers are counted, and a scoring function, learnt with a ranking Perceptron, is applied. This method can be applied to 44% of the TREC02 factoid questions, yielding an improvement of 17.8% in Mean Average Precision (MAP) when adding the semantic graphs to the original surface features. Only Passage Retrieval was evaluated.

In recent years, many researchers have opted for syntax (particularly dependency parsing) as a good representation for the QA problem. This is an intermediate approach between bag-of-words and semantic techniques that can be easily carried out with statistical models. The same kind of techniques have been used for the task of textual entailment [Wang and Neumann, 2007a], which is closely related to Answer Extraction. Basically, syntactic parsing is used on questions and answers to obtain information that is more general and more meaningful at the same time. A statistical correlation between the tree representing the question and the tree surrounding the answer candidate is estimated to help the Answer Extraction. This syntactic comparison has been approached several different ways:

One of the first proposals for using syntactic information for QA is the one of Lin and Pantel [2001]. They use the Minipar dependency parser on the data and learn rules of equivalence between syntactic structures, these rules are later used to find candidate answers as paraphrases of the question. Lin and Pantel evaluate only the quality of the rules, not their effect on QA. Minipar is also used by the PiQASso system [Attardi et al., 2001]. PiQASso uses many linguistic filters, among others checking how many dependency relations from the question can be found in the paragraph surrounding the answer. Unfortunately, they don't evaluate the exact contribution of dependency relation filtering.

Echihabi and Marcu [2003] proposed to build a statistical machine translation (SMT) model to translate the parsing tree of a question to the parsing tree of an answer (including the recognition of the answer string). The parsing trees are modified to make them more abstract (e.g. reduce lexicalization in favour of syntactic classes), and then the SMT model is trained using the IBM Model 4. The corpus of sentences is extracted from the TREC00–01 data and contains 18,618 question/answer pairs. They improved the performance on the TREC02 factoid questions by 11.6%.

Extending the previous research, Cui et al. [2005] learnt an SMT model to translate dependency parsing chains between every pair of words from a question and answer. They reported an improvement of +84% over the pure IR baseline for the passage retrieval sub-task of TREC03.

A simpler method of matching syntactic structures is the one proposed by Tanev et al. [2004]. They use a dependency parser on the question and the retrieved passages. The question is heuristically converted into its affirmative form with the expected position of the answer, and then the whole structure is matched against the candidate sentences. The matched sub-trees are weighted according to a formula involving the `idf` of the subtree words. Tanev et al. report an improvement of 4.6% in Accuracy when using this dependency parsing on the TREC04 corpus.

Moschitti and Quarteroni [2010] took the previous approach one step further. After processing the question and the candidate answer sentence with the Charniak parser and a semantic role labelling system, they used tree-kernel methods in SVMs to compare all possible sub-trees of both sentences and obtain a measure of their similarity. The original ranking based on a bag-of-words approach is then updated with the SVM classification. They trained the reranking model from the TREC01 QA corpus, which is much larger than our EPPS data-sets. Moschitti and Quarteroni report an improvement of 12.8% in MRR on the TREC01 data-set.

Similarly, Shen and Klakow [2006] implemented a reranker based on the correlation of syntactic paths (with several approximated matching strategies) on top of a maximum entropy ranker. They tuned their system using the TREC00–03 data-sets (over 2,300 questions), and tested it with the 2004 edition. They found an improvement of 19.7% in MRR over their density-based baseline extractor for factoid questions.[6]

Bouma et al. [2005] uses also syntactic paths but break them into triplets, i.e. *word1-label–word2*. Then they count the overlap of triplets obtained from question and from candidate answer, and together with other surface measures produce a confidence score for the extraction. They have also developed a set of rules to discover equivalences between syntactic structures (e.g., apposition, coordination, possessive relations, etc.) that boost the recall. They report improvements on the Dutch QA evaluations of CLEF 2003 (11.1%) and 2004 (16.6%).

Moriceau and Tannier [2010] used a similar strategy for factoid questions, employing a dependency parser to process the question, and a set of manually developed transformation rules to convert the parse trees into predicates (e.g. SUBJ(build, Eiffel Tower)) and identify the expected syntactic role of the answer. This expected role is searched in the retrieved passages, and then its support is validated by matching the question predicates in the answer sentence. A final ranking is carried out via a cascade of heuristic scores. Moriceau and Tannier evaluated their method with CLEF and Quaero[7] data, and reported a variable effect: depending on the document collection, this syntactic validation yields improvements of up to 12% and decrements of 1%.

In summary, all of the aforementioned studies bridge the lexical mismatch by considering the syntax of the whole question and comparing it to the whole syntax of a single corpus sentence, as we do with our Syntactic Reranker. They report improvements over the bag-of-words approach ranging from 11% to 19% for different measures and factoid QA test corpora. But all of these approaches have only been designed and evaluated for standard, well-formed written text.[8] As an annotated corpus suitable for training semantic role labellers or parsers for spoken documents does not exist, spoken documents could render these approaches infeasible (i.e. syntactic parsers suffer even when used with out-of-domain documents). In the previous Section, we have evaluated our Syntactic Reranker with a collection of spoken documents. In the manual reference transcripts (the most similar scenario), this improved the baseline MRR by 22.8% or 41.6%, depending on the training set. The relevant features of our approach are that it does not take into account the complete syntactic tree surrounding the candidate answer, but only the relations to the selected keywords. Our approach can capture long dependencies between words while considering surface measures at the same time, making it appropriate for the spoken domain.

Finally, it is interesting to comment the approach of Aktolga et al. [2011]. They follow very closely our approach, but using a different model. They use dependency parsing and make the same assumptions about syntactic relation paths from candidate answers

[6] These are the questions where their system expects a named entity as an answer, disregarding whether this automatically obtained classification is right or wrong.

[7] `http://www.quaero.org`

[8] With the exception of Moriceau and Tannier [2010], who have worked with a collection of web pages.

to keywords, as we do. They also integrate bag-of-words with syntactic techniques in a reranker scheme. The main difference with our extractor is that Aktolga et al. [2011] use a scoring mechanism similar to the IBM Model 1 and statistically estimate the parameters, while we use a machine learning approach. IBM Model 1 considers all possible alignments of label pairs in its estimation. Thus, the structural information given by the label order in the path is not taken into account while our method captures it in the M_k measure. Aktolga et al. [2011] report an improvement of 35% in MRR@5 and 25.6% on accuracy when using the passage reranker. They do not have an Answer Extraction module and evaluate only passage ranking. Thus, when a sentence has several named entities still has to be decided which one is the correct answer. This decision may introduce new errors to the process. In fact, their passage reranker is not much different from any answer extractor discussed in this section, although the results are difficult to compare and probably higher.

8. Coreference Resolution

The Answer Extractor reranker presented in Chapter 7 fulfils the objective of “bridging the lexical chasm” [Berger et al., 2000] but still has important limitations. It compares the syntactic structure of the question to only one single corpus sentence at a time. Thus, if two different keywords are located in two adjacent sentences, but only one sentence has a candidate answer, only the syntactic relations from one sentence will be used. It would be desirable to have a method to bridge this chasm between sentences. Our heuristic surface measures of distance and redundancy are useful because some of them ignore sentence boundaries. However, this is a very crude way of broadening the answer context.

Moriceau and Tannier [2010] identified and addressed this context problem by validating the dependencies extracted from the question in multiple passages or documents. This is implemented as a reranker after the initial set of candidate answers has been gathered. It is used to obtain the text that supports the answer, and this text can come from different sentences. The works of [Moldovan et al., 2007b,a] are implicitly bridging all of a document's semantic information by converting the sentences into logic forms and taking all of them as hypotheses for an automatic theorem prover.

> $\langle \textit{I} \rangle_{\mathfrak{a}}$ *have been visiting* $\langle \textit{Bosnia} \rangle_{\mathfrak{b}}$ *for* $\langle \textit{several years} \rangle_{\mathfrak{c}}$ *and* $\langle \textit{I} \rangle_{\mathfrak{a}}$ *can personally attest to the transformation which has come over* $\langle \textit{the country} \rangle_{\mathfrak{b}}$ *during* $\langle \textit{that period} \rangle_{\mathfrak{c}}$

Figure 8.1: Example of coreference resolution

With the goal of incorporating evidence from several sentences at the same time, we have explored the use of coreference resolution. This Chapter presents an extension of Sibyl that uses a coreference resolution module.

8.1 Introducing Coreference in Sibyl

Coreference resolution is the task of identifying expressions in the text (usually called *mentions*) that refer to the same discourse entity. Figure 8.1 shows a sentence extracted from the EPPS corpus in which entities and mentions have been detected and coreference relations have been labelled. An anaphoric mention is one that cannot be correctly interpreted without knowing what entity it is referring to. In the example presented in Figure 8.1, "the country" and "that period" are anaphoric mentions. Pronouns, too, are always anaphoric.

Some studies have shown that the impact of coreference resolution on standard QA depends mainly on the specific document collection characteristics [Vicedo and Ferrández, 2006]. The larger and more redundant the collection, the smaller the effect of coreference (e.g., a collection of news aggregated from different newspapers for the same time period). This is because the same information is reported several times in the documents and is written in different forms each time. Having this variety and redundancy makes things easier for QA systems based on statistical or pattern-matching techniques. Coreference resolution performs a similar information replication by discovering new occurrences of the text entities.

There are two motivations to perform coreference resolution in the QAst task:

- The QAst collection is very small, and each topic is discussed for a short period; thus, most of the answers are explicitly mentioned very few times. It is a low-redundancy collection, so coreference should increase our recall by helping to detect more candidate answers.
- A method for adding syntactic information to the answer extraction process transforming it into a matching of syntactic sequences has been presented in Section 7.3. Unfortunately, this method has some strong limitations. For example, it is only able to find relations between keywords and candidate answers that are actually in the

same sentence whereas, for instance, the less sophisticated keyword density measure works with a broader sense of proximity across sentences. Using coreference should help identify additional mentions of keywords and candidate answers that may be linked together in new sentences, thus increasing the potential applicability of the Syntactic Reranker to more questions.

We have used RelaxCor, a state-of-the-art coreference resolution system [Sapena et al., 2010], to process each document in the collections. The system first identifies possible mentions and then creates coreference chains that link some of them together. Therefore, each mention in the chain is a reference, anaphoric or not, to the same real-world concept. RelaxCor is a complex natural language processing tool that has been designed for written text and trained with written text documents from the SemEval coreference evaluation campaign.[1]

RelaxCor allows embedded phrases and tackles nominal coreference, i.e., it can disambiguate pronoun referents and identify coreferent noun phrases (including named entities). Mentions are identified using the dependency parser from Section 7.3; all noun phrases are considered mentions. RelaxCor tries to resolve all coreference links in a document at once: it considers every mention as a node in a graph, and then it uses relaxation labeling to find the best graph partition in which every subgraph is composed only of corefering mentions.

We have devised tow ways of using the coreference chains for QA. They can be used by two different modules of our system:

1. *Passage Retrieval*: Once the chosen question keywords have been found in the documents, all the terms corefering with them are marked as occurrences of the keywords, too. Thus, the number of keywords found in each retrieved document is increased. After this expansion of keywords, passages are constructed as shown in Chapter 5, considering both the original keywords and the corefering terms. Finally, the answer extraction process is conducted as usual. By doing so, the number and length of retrieved passages increases and, therefore the number of potential candidate answers that can be found in the passages is increased.

2. *Answer Extraction*: We start with the pool of candidate answers that has been gathered from the named entities (these matching the type of the expected answer type). Then, any term in the passages that is coreferent with a candidate is added to the pool. These new candidates are treated as mentions of the original named entities that are linked to. Notice that pronouns and general noun phrases are not named entities, thus, answering a factoid question with "*they*" or "*that day*" is incorrect. For this reason we keep track of what named entity is the original candidate for each coreference chain. Finally, the expanded pool of candidates is rated with our Answer Extraction techniques and each named entity is assigned the score of its best rated mention. Using this process, the number of candidate answers is not increased, but the number of contexts (i.e. sentences) where they can be evaluated by our Answer Extractor do is increased.

[1] http://semeval2.fbk.eu/

We have implemented and experimented with both ways of incorporating coreference resolution into Sibyl.

When using coreference chains in the Answer Extraction, we can link toghether a mix of anaphoric mentions of candidate answers, anaphoric mentions of keywords, and the original keywords. Thus, they increase the applicability of the syntactic similarity (that is limited to only one sentence), and bridge together all sentences containing information about the same entities. Here, we attempt to explicitly take into account the information contained in all the sentences that hold coreferent mentions of the candidate answer. This is achieved through the features of the reranking models using this process:

- Each candidate answer A is in a coreference chain L, and each coreference chain is composed of a list of corefering mentions, $L = \{a_1, a_2, \ldots a_n\}$. For each different coreference chain a set of features is calculated.

  Each feature described in 7.2 and 7.3, f_i, is added to the set. The value of the feature is selected from the values that this feature obtains with the mentions in the chain. Thus, $f_i(L) = \arg\max_n f_i(a_n)$.

  By using this method, we are gathering evidences supporting the candidate A derived from its n different mentions found in the document. Ideally, some mentions will have high values for complementary features; therefore, there is no need that a single mention has very high values for all the features,

  What is the *highest* value for a feature depends on the meaning of the feature. Usually the desired values are either the smallest ones (e.g., the smallest *Distance from QFW*), or the largest one (e.g., the largest M_k value). Therefore, for each feature, the choosen value is the one that better supports A as a correct answer.

When the features have been calculated, they are assigned to the candidate answer mention that was originaly detected as a suitable named entity. The process of answer extraction and reranking has no further modifications beyond this point.

8.2 Evaluation and Discussion

We have conducted experiments with coreference resolution in two different modules. In the first, coreference is used during Passage Retrieval to obtain words referring to the found keywords, thus expanding the number and length of found passages. The second is used during Answer Extraction to obtain new candidate answers that corefer with the original ones (the original named entities found by our NERC module).

We have only experimented with coreference on manual transcripts due to the limited adaptability of our coreference solver (it has a minimal performance with ASR transcripts). The results achieved with coreference in combination with each reranker are shown in Table 8.1. Columns Top1, Top5 and MRR additionally show the difference between this result and the original result from Table 7.3 (without using coreference). The

Coref. in PR	Coref. in AE	System Name	Top1	Top5	MRR	Accuracy
yes	no	Heuristic Baseline	15 $_{-1}$	33 $_{-1}$	0.2920 $_{-0.0078}$	20.00%
		Heuristic Rerank	18 $_{-1}$	37 $_{+3}$	0.3431 $_{+0.0071}$	24.00%
		Syntactic Rerank	22 $_{=}$	36 $_{-1}$	0.3689 $_{+0.0002}$	29.33%
no	yes	Heuristic Baseline	12 $_{-4}$	32 $_{-2}$	0.2591 $_{-0.0411}$	16.00%
		Heuristic Rerank	18 $_{-1}$	37 $_{+3}$	0.3329 $_{-0.0031}$	24.00%
		Syntactic Rerank	19 $_{-3}$	38 $_{+1}$	0.3447 $_{-0.0240}$	25.33%
yes	yes	Heuristic Baseline	13 $_{-3}$	31 $_{-3}$	0.2653 $_{-0.0349}$	17.33%
		Heuristic Rerank	15 $_{-4}$	36 $_{+2}$	0.3142 $_{-0.0218}$	20.00%
		Syntactic Rerank	18 $_{-4}$	35 $_{-2}$	0.3289 $_{-0.0398}$	24.00%

Table 8.1: Results from Sibyl when using coreference resolution. Differences with results from Table 7.3 are shown in subscript

Coref. in AE	Coref. in PR	Model	Top1	Top5	MRR	Accuracy
yes	no	Heuristic Baseline	14 $_{-2}$	34 $_{=}$	0.2880 $_{-0.0122}$	18.67%
		Heuristic Rerank	18 $_{-1}$	35 $_{+1}$	0.3322 $_{-0.0038}$	24.00%
		Syntactic Rerank	22 $_{=}$	37 $_{=}$	0.3687 $_{=}$	29.33%
no	yes	Heuristic Baseline	12 $_{-4}$	32 $_{+2}$	0.2591 $_{-0.0411}$	16.00%
		Heuristic Rerank	19 $_{=}$	33 $_{-1}$	0.3304 $_{-0.0056}$	25.33%
		Syntactic Rerank	19 $_{-3}$	38 $_{+1}$	0.3447 $_{-0.0240}$	25.33%
yes	yes	Heuristic Baseline	12 $_{-4}$	32 $_{-2}$	0.2591 $_{-0.0411}$	16.00%
		Heuristic Rerank	18 $_{-1}$	37 $_{+3}$	0.3329 $_{-0.0031}$	24.00%
		Syntactic Rerank	19 $_{-3}$	38 $_{+1}$	0.3447 $_{-0.0240}$	25.33%

Table 8.2: Results from Sibyl when using only pronominal coreference. Differences from results in Table 7.3 are shown in subscript

numbers demonstrate that when used in Passage Retrieval, coreference has almost no effect. When coreference is used with Answer Extraction, it mostly harms precision (Top1). In both cases it may increase the coverage (Top5), but not enough to raise MRR. The combination of both modules yields mixed results. It can be recognised that the Heuristic Reranker consistently benefits more from coreference than the Syntactic Reranker. Table 8.2 shows the results from repeating the same experiments, but solving only pronominal coreference instead of any type of coreference. With this setting, coreference still has a negative effect on the Top1 and MRR measures, but may increase Top5 in half of the experiments. Coreference may increase the coverage of the system, but negatively affects the ranking.

Vicedo and Ferrández [2006] report cases of either positive or negative small effects of coreference resolution in QA on written text in their work. Coreference resolution affects MRR in a range from +2.1% to -13.6%, depending on the particular experiment. Our initial hypothesis, that the EPPS corpus has very low redundancy and so coreference should increase the recall and help detect more keywords and candidate answers, is difficult to prove given the results. Coreference helps to slightly increase the recall in many experiments (some new answers are ranked in the Top5), but it is so harmful to precision that the global measures decrease. The reason for this is that coreference adds more noise than information to the process; part of this noise is formed of plainly useless coreferences, and part is formed of false coreferences. Coreference chains are generated by a linguistic analyser designed for written text. Although there is no quantitative evaluation of the chains, it is expected that they are mostly erroneous and introduce false and inconsistent information to the system. To fully understand the effect of coreference in these corpora, we performed a qualitative analysis by hand.

We have evaluated coreference when it is used in the Answer Extraction module to gather mentions of the initial set of candidate answers (i.e. named entities). The objective of using coreference here is to identify anaphoric mentions of the candidates that are not named entities themselves, and are thus not considered as candidate answers for our system. These mentions may either be in a better position for the heuristic measures of redundancy or have more similar syntactic relations (as seen in Section 7.3) than the original ones. These candidate mentions are introducing new information to the system (mentions of entities) that cannot be obtained by a NERC system alone.

The evaluation is as follows: From questions where the correct answer was pooled into the candidate answers set, these correct candidates were selected and manyally inspected by human experts in the search of two different aspects. The first aspect is which words are in the same coreference chains (if any) and what relation do they have to the original candidates. The relations were labelled as one of these 5 classes:

- Orthographic identity: occurrences of the same entity written in the same way.
- Subsumption: the mention is a shorter or longer form of the same name. The words from one of the mentions are a strict subset of the other ones.
- Pronoun: pronoun anaphorically referring to an entity.
- Alias: the mention is a different name that designs the same entity e.g., "*Mister Diamandouros*" referred to as the "*Ombudsman*"."
- Confusion: erroneous linkage of two mentions referring to different entities. This has the effect of giving the same score for all candidate answers in both coreference chains.

Only pronouns and aliases provide the kind of new information that the named entity recogniser cannot capture, yielding mentions to the candidate answers that are not named entities themselves.

The second aspect is to check the cases in which the coreference chains contain mentions not labelled as named entities (i.e. useful information). This checking was done disregarding the correctness of the linkage to the particular mention. Mentions in the Orthographic Identity or Subsumption classes should not provide new useful information, as these mentions can be detected as named entities by our named entity recogniser, thus being added to the candidate answer pool. However, they do provide new information whenever one of the mentions has not been correctly detected by the NERC system,

	Chain	In question
Chains Found	69	25
Good chains	13	10
Wrong chains	48	18
Both Good&Wrong	6	7
Redundant chains	16	11
New information	48	21

Table 8.3: Evaluation of the coreference chains for correct candidate answers. Syntactic Reranker on manual transcripts

	Original	Coref. in AE	New	Lost	Up	Down
Top1	22	19	3	6	–	–
Top5	37	38	3	2	5	7

Table 8.4: Effect of adding coreference to Answer Extraction in the Syntactic Reranker for manual transcripts per question

thus being introduced as a new context where the original entity appears. Even if it is orthographically equal to another entity, redundant links may provide the same new information as pronouns and aliases in a coreference chain.

It must be noted that we have only evaluated coreference chains for *correct* candidate answers. There are hundreds of other candidate answers that are not correct answers and belong to hundreds more coreference chains that also introduce new information to the system. Disregarding the correctness of these coreference chains, their effect on our answer extractor can only be negative given how coreference is handled (i.e. selecting the best value among all mentions in the chain).

Moreover, only factoid questions with non-nil answers where the correct answer was retrieved have been evaluated. This gives a total of 51 questions analysed. Of these 51 questions, only 25 have coreference chains involving any correct candidate answer, totalling 69 different coreference chains. This is an average of 2.76 coreference chains per question. Table 8.3 summarises our analysis. The left column shows the results on the individual coreference chain evaluation, and the right column groups them by question. *Good chains* refers to the number of chains that link either a correct Pronoun or Alias. *Wrong chains* counts the number of chains with Confusion links. *Redundant chains* is the number of chains with only Orthographic identity or Subsumption links, thus adding only redundant information. Finally, *New information* is the number of chains that include new information in the form of mentions not labelled as named entities by our NERC module (either correct or not), as described previously in this Section. From these numbers, we see that 21 out of 25 questions have new information although only 10 have any theoretically *good* coreference chains.

Table 8.4 shows the effect on in Top1 and Top5 when coreference is added to the Answer Extraction module of the Syntactic Reranker. It shows how many questions are

	Total	MRR Increase	MRR Decrease	No change
Total Questions	51	8	9	34
Questions with chains	25	8	9	8
∃ Good chains	10	3	2	5
∃ Wrong chains	18	5	4	9
only Redundant chains	2	1	0	1
Adds new information	21	6	3	12
No new information	30	2	6	21

Table 8.5: Effect of adding coreference to AE on the Syntactic Reranker for manual transcripts per chain type

Mention 1: *the overall development of our by and large excellent relations with* [*China*]$_{\mathrm{ORG}}$ *and the arms embargo debate naturally plays out in that context*

Mention 2: *the further development of bilateral relations*

Figure 8.2: Example of coreference resolution

newly answered in the Top5, how many lose all correct answers, and how many questions have correct answers moving up or down within the top 5 positions. Although the numbers are very low, the inspection of this table shows that coreference is harmful for Top1 and does not improve Top5.

Table 8.5 has a more detailed evaluation in terms of MRR score. It reports how many questions increase or decrease their particular MRR score when coreference is added. In the second block of rows, the 25 questions are classified according to the presence of Good chains, Wrong chains, or only Redundant chains (some questions have both Good and Wrong chains). This reveals that all three types have an almost negligible effect, as increments are balanced with decrements for all three types. In the third block of rows, the questions are split according to whether they have new information or only redundant information. In this case, we can see a clearer pattern: coreference chains that add new information have a positive effect on Answer Extraction, whereas the others have a negative effect. This allows us to extend the conclusions of Vicedo and Ferrández [2006]: in low redundancy corpora, coreference is not useful if it only detects trivial coreferences. For our approach, coreference must be able to enrich the pool of candidate answers with new information, not just new links between candidates. Unfortunately, the amount of data in the EPPS transcripts is too small to quantify our conclusions.

Our evaluation also sheds light on the poor performance of RelaxCor on this corpus. A very frequently observed phenomenon is the detection of too many unreliable mentions. Figure 8.2 shows two noun phrases that are considered coreferent, but this relation is incorrect because Mention1 is incorrectly detected as a mention and should not have been linked to any other entity in the text. Although this relation is not completely redundant, as it adds new information to our Answer Extractor by linking "*China*" with a mention encompassing only entities that are unsuitable as candidate answers, thus mainly adding undesired noise.

9. Conclusions

In this dissertation, we have detailed our work on the topic of factoid question answering for spoken documents. The main contributions of this work are:

- The development of a modular and flexible factoid Question Answering system tailored for handling speech transcripts named Sibyl.

  Sibyl takes advantage of several natural language analysers, incorporating linguistic information from named entities, syntactic dependencies, and coreference chains. All this information is obtained with machine learning–based tools. As a consequence, Sibyl could be adapted to other domains, or even other languages, following the same architecture provided there was an adequate availability of tools and annotated resources. These tools were originally developed for written text, being its adaption in the spoken setting a major challenge. We have adapted and enriched some of these, and have studied which state-of-the-art techniques perform better for speech transcripts.

- We have impulsed the creation of an evaluation framework for question answering on spoken documents. This framework, called QAst, contains several evaluation scenarios featuring different kinds of speech (i.e., meeting, seminar, politic speech and broadcast news), is available in three languages (i.e., English, Spanish and French), and different kinds of questions (i.e., written questions and spontaneous oral questions). This evaluation framework has helped the creation of literature on question answering on spoken documents and to settle this as a stand-alone research topic.

Sibyl QA system has been extensively evaluated on the QAst evaluation framework. We have analysed the effect of the QA pipeline modules one by one, and evaluated the usefulness of several kinds of linguistic information within the QA process. The main conclusions drawn from the experimental study are as follows:

- Regarding the problem of Spoken Document Retrieval, we have shown a novel method that can overcome part of automatic speech recognition errors using a sound measure of phonetic similarity based on phonetic sequence alignment. It can be used in combination also with traditional document ranking models. This method is totally independant of the ASR and can be applied on any kind of transcription and any language.
- We have presented a simple Answer Extraction module based on shallow heuristic measures. We also show that this module can be significantly improved by using machine learning and syntactic information in a reranking scheme, especially in the case of manual transcripts. Remarkably, this improvement is achieved by using a very small training set with examples from only 50 questions. It is worth noting this proves that:
  - Dependency parsing is able to provide reliable information and improve the overall results when working on relatively good automatic ASR transcripts. Where "relatively" is estimated as having a WER $< 25\%$ in our experiments.
  - The size of the development corpus has a critical effect in the answer ranking learning. We have showed that using a three times larger training set (by doing a leave-one-out evaluation using all available data) overall results can be improved by about 6 MRR and sccuracy points in the case of manual transcripts, being this performance quite close to the actual upper bound of our Answer Extractor.
- We have presented a NERC module especially enriched to work with automatic transcripts. Its performance is quite modest (below 60% for automatic transcripts), but, very interestingly, the negative impact on the final results is minor. The number of correctly tagged answers decreases only an 8% in average with respect to the answers contained in the retrieved passages of automatic transcripts.
- We presente several experiment on the use of coreference resolution in QA. The results show that the use of coreference resolution might help to increase the coverage of possible answer candidates from automatic transcripts, but the negative effect on the precision is larger, making the overall performance to generally decrease.
- Overall, the results of Sibyl in the QAst evaluation scenario are comparable or better in some cases than the state-of-the-art systems. This is remarkable, since Sibyl relies

solely on information automatically learnt from examples. We also show that the amount of examples is critial regarding the performance. In the same line, SIBYL has presented a higher robustness when moving from correct manual transcripts to the automatic transcripts generated by ASRs.

We believe that with the current NLP technology more in-depth approaches to QA would be infeasible when dealing with spoken documents. Further work will be devoted to the task of more analysis and evaluation of the linguistic processors on spoken documents. We think there's still room for improving the NERC quality in automatic transcripts, specially introducing contextual measures of ASR confidence. Creating new evaluation question sets would be a helpful task for both the development of SIBYL and for the research community.

Bibliography

T. Akiba and H. Tsujimura. Error-tolerant question answering for spoken documents. *Proceedings of the INTERSPEECH 2007 Conference*, 2007.

E. Aktolga, J. Allan, and D. Smith. Passage reranking for question answering using syntactic structures and answer types. *Advances in Information Retrieval*, 2011.

S. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215, 1990.

M. Alzghool and D. Inkpen. University of Ottawa's participation in the CL-SR task at CLEF 2006. *Proceedings of the CLEF 2006 Workshop*, 2006.

G. Amati and C. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4), 2002. ISSN 1046-8188.

I. Androutsopoulos, G. Ritchie, and P. Thanisch. Natural language interfaces to databases, an introduction. *Natural Language Engineering*, 1(01):29–81, 1995.

G. Attardi, A. Cisternino, F. Formica, M. Simi, A. Tommasi, and C. Zavattari. PiQASso: Pisa question answering system. In *Proceedings of Text REtrieval Conference (TREC)*, 2001.

S. Babych, A. Henn, J. Pawellek, and S. Padó. Dependency-based answer validation for german. *Working Notes of the CLEF 2011 Workshop*, 2011.

F. Batista, D. Caseiro, N. Mamede, and I. Trancoso. Recovering capitalization and punctuation marks for automatic speech recognition: Case study for portuguese broadcast news. *Speech Communication*, 50(10), 2008.

A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the SIGIR Conference*. ACM, 2000.

G. Bernard, S. Rosset, O. Galibert, G. Adda, and E. Bilinski. The LIMSI participation in the QAst 2009 track: experimenting on answer scoring. In *Proceedings of CLEF 2009 Workshop*. Springer-Verlag, 2009.

G. Bernard, S. Rosset, M. Adda-Decker, and O. Galibert. A question-answer distance measure to investigate QA system progress. In *Proceedings of the LREC 2010 Conference*, 2010.

M. Bilotti, J. Elsas, J. Carbonell, and E. Nyberg. Rank learning for factoid question answering with linguistic and semantic constraints. In *Proceedings of the CIKM Conference*. ACM, 2010.

J. Bos and M. Nissim. Cross-lingual question answering by answer translation. In *Workshop of Cross-Language Evaluation Forum (CLEF)*, 2006.

G. Bouma, J. Mur, and G. van Noord. Reasoning over dependency relations for QA. In *Proceedings of the IJCAI workshop on Knowledge and Reasoning for Answering Questions (KRAQ)*, pages 15–21, 2005.

T. Brants. TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000.

E. Brill, S. Dumais, and M. Banko. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 257–264. Association for Computational Linguistics, 2002.

C. Brun and M. Ehrmann. Adaptation of a named entity recognition system for the ester 2 evaluation campaign. In *Proceedings of the NLP-KE Conference*. IEEE, 2009.

J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrinhari, T. Strzalkowski, E. Voorhees, and R. Weischedel. Issues, tasks and program structures to roadmap research in question & answering (Q&A). *NIST*, 2001.

D. Buscaldi, P. Rosso, J. Turmo, and P. R. Comas. Towards the evaluation of voice-activated question answering systems: Spontaneous questions for QAst 2009. In *Proceedings of the III Jornadas PLN-TIMM.*, 2009.

E. Cabrio, M. Kouylekov, B. Magnini, M. Negri, L. Hasler, C. Orasan, D. Tomás, J. Vicedo, G. Neumann, and C. Weber. The QALL-ME benchmark: a multilingual resource of annotated spoken requests for question answering. *Proceedings of the Sixth International Language Resources and Evaluation (LREC 08), Marrakech*, 2008.

Y. Chali and S. Dubien. University of Lethbridge's participation in TREC-2007 QA track. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC-2007)*, 2007.

Y. Chali, S. Joty, and S. Hasan. Complex question answering: unsupervised learning approaches and experiments. *Journal of Artificial Intelligence Research*, 35(1):1, 2009.

J. Chu-Carroll and J. Fan. Leveraging wikipedia characteristics for search and candidate generation in question answering. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

J. Chu-Carroll, K. Czuba, J. Prager, A. Ittycheriah, and S. Blair-Goldensohn. IBM's PI-QUANT II in TREC 2004. In *Proceedings or TREC conference*, 2004.

C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilker. Statistical selection of exact answers (MultiText experiments for TREC 2002). In *Proceedings of the Eleventh Text REtrieval Conference (TREC-2002)*, 2002.

P. R. Comas and J. Turmo. Spoken document retrieval based on approximated sequence alignment. *11th International Conference on Text, Speech and Dialogue (TSD 2008)*, 2008a.

P. R. Comas and J. Turmo. Robust question answering for speech transcripts: UPC experience in QAst 2008. *Proceedings of the CLEF 2008 Workshop*, 2008b.

P. R. Comas and J. Turmo. Robust question answering for speech transcripts: UPC experience in QAst 2009. In *Proceedings of the CLEF*, Berlin, Heidelberg, 2009. Springer-Verlag.

P. R. Comas, J. Turmo, and M. Surdeanu. Robust question answering for speech transcripts using minimal syntactic analysis. *Proceedings of the CLEF 2007 Workshop*, 2007.

P. R. Comas, J. Turmo, and L. Màrquez. Using dependency parsing and machine learning for factoid question answering on spoken documents. In *Proceedings of the INTERSPEECH 2010 Conference*, Makuhari, Japan, September 2010.

A. Copestake and K. Spärck-Jones. Natural language interfaces to databases. *Knowledge Engineering Review*, 5(4):225–249, 1989.

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

H. Cui, R. Sun, K. Li, M. Kan, and T. Chua. Question answering passage retrieval using dependency relations. In *Proceedings of the SIGIR Conference*. ACM, 2005.

T. Dalmas and B. Webber. Answer comparison in automated question answering. *Journal of Applied Logic*, 5(1), 2007.

H. Dang and K. Owczarzak. Overview of the tac 2008 opinion question answering and summarization tasks. In *Proc. of the First Text Analysis Conference*, 2008.

H. T. Dang, D. Kelly, and J. Lin. Overview of the TREC 2007 question answering track. *The Sixteen Text Retrieval Conference (TREC 2007) Proceedings*, 2007.

I. Dornescu. Semantic QA for encyclopaedic questions: EQUAL in GikiCLEF. *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, 2010.

S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM, 2002.

A. Echihabi and D. Marcu. A noisy-channel approach to question answering. In *Proceedings of the ACL Conference*. Association for Computational Linguistics, 2003.

C. España-Bonet and P. R. Comas. Full machine translation for factoid question answering. In *Proceedings of the EACL workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT)*, Avignon, France, April 2012. Association for Computational Linguistics.

J. Fan, A. Kalyanpur, J. Murdock, and B. Boguraev. Mining knowledge from large corpora for type coercion in question answering. *The Semantic Web-ISWC*, 2011.

B. Favre, F. Béchet, and P. Nocéra. Robust named entity extraction from large spoken archives. In *Proceedings of the HLT-EMNLP Conference*. ACL, 2005.

J. Fayolle, F. Moreau, C. Raymond, and G. Gravier. Reshaping automatic speech transcripts for robust high-level spoken document analysis. In *Proceedings of the AND Workshop*. ACM, 2010.

D. Ferrucci, E. Nyberg, J. Allan, K. Barker, E. Brown, J. Chu-Carroll, A. Ciccolo, P. Duboue, J. Fan, D. Gondek, E. Hovy, B. Katz, A. Lally, M. McCord, P. Morarescu, B. Murdock, B. Porter, J. Prager, T. Strzalkowski, C. Welty, and W. Zadrozny. Towards the open advancement of question answering systems. *IBM Research Report. RC24789 (W0904-093), IBM Research, New York*, 2009.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

U. Furbach, I. Glöckner, and B. Pelzer. An application of automated reasoning in natural language question answering. *AI Communications*, 23(2), 2010.

S. Galliano, E. Geoffrois, G. Gravier, J. Bonastre, D. Mostefa, and K. Choukri. Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *Proceedings of LREC*, page 315–320, 2006.

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.

E. Gonzàlez. Una nova guia per al MetaServer. *LSI Internal Report: LSI-09-1-R (UPC)*, 2009.

D. Graff. *The AQUAINT corpus of English news text*. Linguistic Data Consortium, 2002.

B. Green, A. Wolf, C. Chomsky, and K. Laughery. *Baseball: an automatic question-answerer*. Massachusetts Institute of Technology, Lincoln Laboratory, 1961.

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.

S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the ACL Conference*, 2006.

S. Harabagiu, D. Moldovan, and J. Picone. Open-domain voice-activated question answering. *Proceedings of the COLING Conference*, 2002.

S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang. Employing two question answering systems in trec-2005. In *Proceedings of the fourteenth text retrieval conference (TREC-14)*, 2005.

S. Harabagiu, F. Lacatusu, and A. Hickl. Answering complex questions with random walk models. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006.

S. Hartrumpf, I. Glöckner, and J. Leveling. Efficient question answering with question decomposition and multiple answer streams. *Evaluating Systems for Multilingual and Multimodal Information Access*, 2009.

M. H. Heie, J. R. Novak, E. W. D. Whittaker, and S. Furui. CLEF 2009 question answering experiments at tokyo institute of technology. In *Proceedings of CLEF 2009 Workshop*. Springer-Verlag, 2009.

U. Hermjakob, A. Echihabi, and D. Marcu. Natural language based reformulation resource and wide exploitation for question answering. In *Proceedings of the eleventh text retrieval conference (TREC 2002)*, 2002.

L. Hirschman and R. Gaizauskas. Natural language question answering: The view from here. *Natural Language Engineering*, 7(04):275–300, 2001.

E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C. Lin. Question answering in webclopedia. *Proceedings of the Tenth Text REtrieval Conference (TREC-2001)*, pages 655–664, 2001.

D. Inkpen, M. Alzghool, and A. Islam. Using various indexing schemes and multiple translations in the CL-SR task at CLEF 2005. *Proceedings of CLEF 2005 Workshop*, 2006a.

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

V. Jijkoun and M. De Rijke. Answer selection in a multi-stream open domain question answering system. *Advances in Information Retrieval*, 2004.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

G. Jones, K. Zhang, and A. Lam-Adesina. Dublin city university at CLEF 2006: Cross-language speech retrieval (CL-SR) experiments. *In Proceedings of the CLEF 2006 Workshop*, 2006.

M. Kaisser. Answer sentence retrieval by matching dependency paths acquired from question/answer sentence pairs. In *Proceedings of the EACL Conference*, Avignon, France, April 2012. Association for Computational Linguistics.

M. Kaisser, S. Scheible, and B. Webber. Experiments at the university of edinburgh for the TREC 2006 QA track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC-2006)*, 2006.

A. Kalyanpur, J. Murdock, J. Fan, and C. Welty. Leveraging community-built knowledge for type coercion in question answering. *The Semantic Web-ISWC*, 2011a.

A. Kalyanpur, S. Patwardhan, B. Boguraev, A. Lally, and J. Chu-Carroll. Fact-based question decomposition for candidate answer re-ranking. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011b.

B. Katz. Annotating the world wide web using natural language. In *In Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet*, 1997.

B. Kessler. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103, 2005.

J. Ko, E. Nyberg, and L. Si. A probabilistic graphical model for joint answer ranking in question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

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*, 2000.

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

C. Kwok, O. Etzioni, and D. Weld. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)*, 19(3):242–262, 2001.

J. Kürsten, H. Kundisch, and M. Eiblu. QA extension for Xtrieval: Contribution to the QAst track. In *Proceedings of the CLEF 2008 Workshop*, 2008.

L. Lamel, G. Adda, E. Bilinski, and J.-L. Gauvain. Transcribing lectures and seminars. *Proceedings of the INTERSPEECH 2005 Conference*, 2005.

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

D. Lin and P. Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4), 2001.

J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 116–123. ACM, 2003.

X. Lluís, S. Bott, and L. Màrquez. A second-order joint eisner model for syntactic and semantic dependency parsing. In *Proceedings of the CoNLL 2009 Shared Task Session*, 2009.

B. Magnini, M. Negri, R. Prevete, and H. Tanev. Mining knowledge from repeated co-occurrences: DIOGENE at TREC-2002. In *Proceedings of the Eleventh Text REtrieval Conference (TREC-2002)*, 2002.

J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, 1999.

A. Mendes and L. Coheur. An approach to answer selection in question-answering based on semantic relations. In *Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

D. Moldovan and V. Rus. Logic form transformation of wordnet and its applicability to question answering. In *Proceedings of the ACL conference*. ACL, 2001.

D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Girju, and V. Rus. LASSO: A tool for surfing the answer net. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 1999.

D. Moldovan, C. Clark, and M. Bowden. Lymba's PowerAnswer 4 in TREC 2007. *The Sixteen Text Retrieval Conference (TREC 2007) Proceedings*, 2007a.

D. Moldovan, C. Clark, S. Harabagiu, and D. Hodges. Cogex: A semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic*, 5(1), 2007b.

D. Mollá and J. Vicedo. Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1), 2007.

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*, 2007.

C. Monz. Minimal span weighting retrieval for question answering. In *Proceedings of the SIGIR Workshop on Information Retrieval for Question Answering*, 2004.

V. Moriceau and X. Tannier. FIDJI: using syntax for validating answers in multiple documents. In *Focused Retrieval and Result Aggregation*, Information Retrieval. Springer, 2010.

A. Moschitti and S. Quarteroni. Linguistic kernels for answer re-ranking in question answering systems. *Information Processing & Management*, 2010.

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*, 2007.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, 2007.

A. Novischi and D. Moldovan. Question answering with lexical chains propagating verb arguments. In *Proceedings of the ACL Conference*. Association for Computational Linguistics, 2006.

D. Oard, J. Wang, G. Jones, R. 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*, 2006.

G. Paaß, A. Pilz, and J. Schwenninger. Named entity recognition of spoken documents using subword units. In *2009 IEEE International Conference on Semantic Computing ICSC'09*. IEEE, 2009.

M. Pardiño, J. M. Gómez, H. Llorens, R. Muñoz-Terol, B. Navarro-Colorado, E. Saquete, P. Martínez-Barco, P. Moreda, and M. Palomar. Adapting IBQAS to work with text transciptions in QAst task: IBQAst. In *Proceedings of the CLEF 2008 Workshop*, 2008.

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

M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz. Sentence segmentation and punctuation recovery for spoken language translation. In *IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2008.*, 2008.

P. Pecina, P. Hoffmannová, G. Jones, Y. Zhang, and D. Oard. Overview of the CLEF-2007 cross-language speech retrieval track. *Proceedings of the CLEF 2007 Workshop*, 2007.

A. Peñas, P. Forner, Á. Rodrigo, R. Sutcliffe, C. Forascu, and C. Mota. Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation. In *Working Notes of the CLEF 2010 Workshop*, 2010.

A. Phillips. *Memo 16 - A question-answering routine*. PhD thesis, Massachusetts Institute of Technology, Dept. of Mathematics, 1960.

J. Prager, E. Brown, A. Coden, and D. Radev. Question answering by predictive annotation. *Proceedings of the SIGIR Conference*, 2000.

J. Prager, J. Chu-Carroll, K. Czuba, C. Welty, A. Ittycheriah, and R. Mahindru. IBM's PIQUANT in TREC2003. In *Proceedings of Text REtrieval Conference (TREC)*, 2003.

V. Punyakanok, D. Roth, and W. Yih. Natural language inference via dependency tree mapping: An application to question answering. *Computational Linguistics*, 2004.

D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

A. Reyes-Barragán, L. Villaseñor-Pineda, and M. Montes-y-Gómez. INAOE at QAst 2009: Evaluating the usefulness of a phonetic codification of transcriptions. In *Proceedings of CLEF 2009 Workshop*. Springer-Verlag, 2009.

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

S. Rosset, O. Galibert, G. Adda, and E. Bilinski. The LIMSI participation in the QAst track. *Proceedings of the CLEF 2007 Workshop*, 2007.

S. Rosset, O. Galibert, G. Bernard, E. Bilinski, and G. Adda. The LIMSI participation to the QAst track. In *Proceedings of the CLEF 2008 Workshop*, 2008.

B. Sacaleanu, G. Neumann, and C. Spurk. DFKI-LT at QA@CLEFST 2007. In *Working Notes for the CLEF 2007 Workshop*, 2007.

G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Cornell University, 1987.

E. Sapena, L. Padró, and J. Turmo. A global relaxation labeling approach to coreference resolution. In *Proceedings of COLING Conference*, Beijing, China, August 2010.

E. Saquete, P. Martínez-Barco, R. Muñoz, and J. Vicedo. Splitting complex temporal questions for question answering systems. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004.

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

N. Schlaefer, J. Chu-Carroll, E. Nyberg, J. Fan, W. Zadrozny, and D. Ferrucci. Statistical source expansion for question answering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011.

D. Shen and D. Klakow. Exploring correlation of dependency relation paths for answer extraction. In *Proceedings of the ACL Conference*. Association for Computational Linguistics, 2006.

D. Shen and M. Lapata. Using semantic roles to improve question answering. In *Proceedings of the EMNLP-CoNLL*, 2007.

D. Sonntag. Introspection and adaptable model integration for dialogue-based question answering. In *Proceedings of the 21st international Jont Conference on Artifical intelligence (IJCAI)*, 2009.

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

S. Stenchikova, D. Hakkani-Tür, and G. Tur. QASR: Question answering using semantic roles for speech interface. *Proceedings of the INTERSPEECH 2006 Conference*, 2006.

R. Sun, J. Jiang, Y. F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. Using syntactic and semantic relation analysis in question answering. In *The Fourteen Text Retrieval Conference (TREC 2005) Proceedings*, 2005.

M. Surdeanu, J. Turmo, and E. Comelles. Named entity recognition from spontaneous open-domain speech. *Proceedings of the INTERSPEECH 2005 Conference*, 2005.

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 INTERSPEECH 2006 Conference*, 2006.

H. Tanev, M. Kouylekov, and B. Magnini. Combining linguistic processing and web mining for question answering: Itc-irst at trec-2004. In *Proceedings of the 13th Text Retrieval Conference (TREC-2004)*, 2004.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL Conference*. Edmonton, Canada, 2003.

J. Turmo, P. Comas, C. Ayache, D. Mostefa, S. Rosset, and L. Lamel. Overview of QAst 2007. *Proceedings of the CLEF 2007 Workshop*, 2007.

J. Turmo, P. Comas, S. Rosset, L. Lamel, N. Moreau, and D. Mostefa. Overview of QAst 2008. *Proceedings of the CLEF 2008 Workshop*, 2008.

J. Turmo, P. Comas, S. Rosset, O. Galibert, N. Moreau, D. Mostefa, P. Rosso, and D. Buscaldi. Overview of QAst 2009. *Proceedings of the CLEF 2009 Workshop*, 2009.

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 INTERSPEECH 2007 Conference*, 2007.

M. Van Zaanen, D. Mollá, and L. Pizzato. AnswerFinder at TREC 2006. *The Fifteenth Text Retrieval Conference (TREC 2006) Proceedings*, 2006.

S. Verberne, L. Boves, N. Oostdijk, and P. Coppen. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

J. Vicedo and A. Ferrández. Coreference in Q & A. In T. Strzalkowski and S. Harabagiu, editors, *Advances in Open Domain Question Answering*, volume 32 of *Text, Speech and Language Technology*. Springer Netherlands, 2006.

J. Vicedo, F. Llopis, and A. Ferrández. University of Alicante experiments at TREC 2002. In *Proceedings of the Eleventh Text REtrieval Conference (TREC-2002)*, 2002.

E. M. Voorhees. Overview of the TREC 2004 question answering track. *The Thirteen Text Retrieval Conference (TREC 2004) Proceedings*, 2004.

J. Wang and D. Oard. CLEF-2005 CL-SR at maryland: Document and query expansion using side collections and thesauri. *Proceedings of the CLEF 2005 Workshop*, 2005.

R. Wang and G. Neumann. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. ACL, 2007a.

R. Wang and G. Neumann. DFKI-LT at AVE 2007: Using recognizing textual entailment for answer validation. *online proceedings of CLEF*, 2007b.

R. Wang, N. Schlaefer, W. Cohen, and E. Nyberg. Automatic set expansion for list question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 947–954. Association for Computational Linguistics, 2008.

I. Weber, A. Ukkonen, and A. Gionis. Answers, not links: extracting tips from yahoo! answers to address how-to web queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012.

E. Whittaker, J. 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.

M. Wu, M. Duan, S. Shaikh, S. Small, and T. Strzalkowski. ILQUA - an IE-driven question answering system. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC-2005)*, 2005.

H. Yang and T. Chua. The integration of lexical knowledge and external resources for question answering. In *The eleventh Text REtrieval Conference (TREC-11)*, 2002.

A. List of Publications

This is a list of the papers that this thesis has produced so far.

- **Design and performance analysis of a factoid question answering system for spontaneous speech transcriptions.**

  Mihai Surdeanu, David Dominguez-Sal, and Pere R. Comas.

  Proceedings of the International Conference on Spoken Language Processing (INTERSPEECH 2006). Pittsburg, September 2006.

  This paper introduces a QA system designed from scratch to handle speech transcriptions. It uses a mix of IR-oriented methodologies and a small number of robust NLP components. We evaluate the system on transcriptions of spontaneous speech from several 1-hour-long seminars and presentations and show that the system obtains a very encouraging performance.

- **Overview of QAst 2007.**

  Jordi Turmo, Pere R. Comas, Christele Ayache, Djamel Mostefa, Sophie Rosset, and Lori Lamel.

  Proceedings of the CLEF 2007 Workshop. Budapest, September 2007.

  This paper describes QAst, a pilot track of CLEF 2007 aimed at evaluating the task of Question Answering in Speech Transcripts. The paper summarises the evaluation framework, the systems that participated and the results achieved.

- **Robust Question Answering for Speech Transcripts Using Minimal Syntactic Analysis.**

  Pere R. Comas, Jordi Turmo and Mihai Surdeanu.

  Proceedings of the CLEF 2007 Workshop on Cross-Language Information Retrieval and Evaluation. Budapest, September 2007.

  This paper describes our participation in the CLEF 2007 Question Answering on Speech Transcripts track. For the processing of manual transcripts we have deployed a robust factoid Question Answering that uses minimal syntactic information. For the handling of automatic transcripts we combine the QA system with a novel Passage Retrieval and Answer Extraction engine, which is based on a sequence alignment algorithm that searches for "sounds like" sequences in the document collection.

- **Question Answering on Speech Transcriptions: the QAST evaluation in CLEF.**

  Lori Lamel, Sophie Rosset, Christelle Ayache, Djamel Mostefa, Jordi Turmo, Pere R. Comas.

  In *Proceedings of 6th Language Resources and Evaluation Conference (LREC 2008)*. Marrakech, May 2008.

  This is a dissemination paper that describes the resources and evaluation framework produced for the firs QAst evaluation.

- **Overview of QAst 2008.**

  Jordi Turmo, Pere R. Comas, Sophie Rosset, Lori Lamel, Nicolas Moreau, and Djamel Mostefa.

  Proceedings of the CLEF 2008 Workshop, Århus, September 2008.

  This paper describes QAst, a pilot track of CLEF 2008 aimed at evaluating the task of Question Answering in Speech Transcripts. The paper summarises the evaluation framework, the systems that participated and the results achieved.

- **Robust question answering for speech transcripts: UPC experience in QAst 2008**.

  Pere R. Comas and Jordi Turmo.

  Proceedings of the CLEF 2008 Workshop on Cross-Language Information Retrieval and Evaluation. Århus, September 2008.

  This paper describes our participation in the CLEF 2008 QAst track. We have participated in the English and Spanish scenarios of QAst.

- **Spoken Document Retrieval Based on Approximated Sequence Alignment.**

  Pere R. Comas and Jordi Turmo.

  Text, Speech and Dialogue, 11th International Conference, TSD. Brno, September 2008.

  In this paper presents a new approach to spoken document retrieval To overcome ASRs limitations, our method is based on an approximated sequence alignment algorithm to search "sounds like" sequences. Our approach outperforms the precision of state-of-the-art techniques in our experiments.

- **Overview of QAst 2009.**

  Jordi Turmo, Pere R. Comas, Sophie Rosset, Olivier Galibert, Nicolas Moreau, Djamel Mostefa, Paolo Rosso, and Davide Buscaldi.

  Proceedings of the CLEF 2009 Workshop on Cross-Language Information Retrieval and Evaluation, Athens, September 2009.

  This paper describes QAst, a pilot track of CLEF 2009 aimed at evaluating the task of Question Answering in Speech Transcripts. The paper summarises the evaluation framework, the systems that participated and the results achieved.

- **Robust question answering for speech transcripts: UPC experience in QAst 2009**.

  Pere R. Comas and Jordi Turmo.

  Proceedings of the CLEF 2009 Workshop on Cross-Language Information Retrieval and Evaluation. Athens, September 2009.

  This paper describes our participation in the CLEF 2009 Question Answering on Speech Transcripts track. We have participated in the English and Spanish scenarios of QAst. For both manual and automatic transcripts we have used a robust factoid Question Answering that uses minimal syntactic information. We have also developed a NERC designed to handle automatic transcripts.

- **Towards the evaluation of voice-activated question answering systems: Spontaneous questions for QAst 2009.**

  Davide Buscaldi, Paolo Rosso, Jordi Turmo, and Pere R. Comas.

  In *Proceedings of the III Jornadas PLN-TIMM.* Madrid, 2009.

  This is a report of the work carried out in order to introduce "spontaneous" questions into QAst at CLEF 2009. The aim of this report is to show how difficult can be to generate "spontaneous" questions and the importance to take into account the real information needs of users for the evaluation.

- **Using dependency parsing and machine learning for factoid question answering on spoken documents.**

  Pere R. Comas, Jordi Turmo, and Lluís Màrquez.

  In *Proceedings of the 13th International Conference on Spoken Language Processing (INTERSPEECH 2010)*, Makuhari, Japan, September 2010.

  In this paper we present two approaches to answer extraction for speech corpora that apply machine learning to improve the coverage and precision of the extraction. The first one is a reranker that uses only lexical information, the second one uses dependency parsing to score robust similarity between syntactic structures. Our experimental results show that a dependency parser can be useful for speech transcripts.

- **Evaluation Protocol and Tools for Question-Answering on Speech Transcripts.**

  Nicolas Moreau, Olivier Hamon, Djamel Mostefa, Sophie Rosset, Olivier Galibert, Lori Lamel, Jordi Turmo, Pere R. Comas, Paolo Rosso, Davide Buscaldi and Khalid Choukri.

  In *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*. La Valleta, May 2010.

  This paper details the evaluation protocol and tools developed for the CLEF-QAst evaluation campaigns that have taken place between 2007 and 2009.

- **Sibyl, a Factoid Question Answering System for Spoken Documents.**

  Pere R. Comas, Jordi Turmo, Lluís Màrquez.

  Submited to the *ACM Transactions On Information Systems (TOIS) Special Issue on Searching Speech* in March 2012. It has undergone a *Major Revision* and a *Minor Revision*.

  This paper gives a detailed presentation of a factoid question answering system, named Sibyl, specifically tailored for QA on spoken documents. As a novelty, Sibyl proposes the usage of several levels of linguistic information for the speech-based QA task, ranging from named entity detection to syntactic parsing and coreference resolution.

- **Full Machine Translation for Factoid Question Answering.**

  Cristina España-Bonet, Pere R. Comas

  In *Proceedings of the EACL Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT)*, Avignon, France, April 2012.

  In this work we present a QA system that uses Statistical Machine Translation for the Answer Extraction part. SMT is used as a means to learn to translate the question into a set of patterns that define the answer context. These patterns are searched in the retrieved documents to extract the exact answer.

Aquesta edició de
Factoid Question Answering for Spoken Documents,
es va acabar d'escriure el dia 23 d'Abril del 2012,
diada de Sant Jordi.
Ha estat editada amb XeLaTeX versió 3.1415926-2.3-0.9997.5 i els tipus
Dax, Optima, *Optima Nova*, Inconsolata, i Gentium,
i Euler per els símbols matemàtics,
i les figures s'han realitzat amb tikz.

⋐◎⋑