

Free University of Bozen-Bolzano  
Faculty of Computer Science

# DIPLOMA THESIS



FREIE UNIVERSITÄT BOZEN  
LIBERA UNIVERSITÀ DI BOLZANO  
FREE UNIVERSITY OF BOZEN · BOLZANO

Zorana Ratkovic

## Deep analysis in IQA: evaluation on real users' dialogues

Knowledge Representation and Databases Research  
Center

Supervisor: asst. prof. Raffaella Bernardi, Ph.D.  
Study program: Computer Science

European Masters Program in Language and  
Communication Technologies (LCT)

2009

Charles University in Prague  
Faculty of Mathematics and Physics

# DIPLOMA THESIS



Zorana Ratkovic

## Deep analysis in IQA: evaluation on real users' dialogues

Institute of Formal and Applied Linguistics

Supervisor: asst. prof. RNDr. Vladislav Kuboň, Ph.D.

Study program: Computer Science

Study specialization: Mathematical Linguistics

European Masters Program in Language and  
Communication Technologies (LCT)

2009

# Acknowledgements

This thesis was completed as a result of my studies at the Charles University in Prague and the Free University of Bozen-Bolzano. I want to thank all the Professors, administrative staff and my fellow students at both of these Universities.

The thesis would not be possible without the help and insight of my two supervisors: Dr. Raffaella Bernardi and Dr. Vladislav Kuboň. Thank you for your understanding and support. I especially want to thank Dr. Bernardi for the countless hours she has spent on this thesis with me. It is through this experience that I have been able to learn and grow as both a student and a person. Further, I want to thank Manuel Kirschner, the PhD student working on Bob at the Free University of Bozen-Bolzano for all his help and hard work on both Bob and my thesis. I also want to thank all the people who have worked on Bob, including the University of Bozen-Bolzano librarians, students and staff - without your work my thesis would not be possible.

Finally, I want to thank my family and friends for their love, support and understanding during these past two years.

# Abstract

Title: Deep analysis in IQA: evaluation on real Users dialogues  
Author: Zorana Ratkovic

Department: KRDB, Free University of Bozen-Bolzano  
Supervisor: asst. prof. Raffaella Bernardi, Ph.D.  
Supervisor's email: [bernardi@inf.unibz.it](mailto:bernardi@inf.unibz.it)

Department: Institute of Applied and Formal Linguistics  
Supervisor: asst. prof. RNDr. Vladislav Kuboň, Ph.D.  
Supervisor's e-mail: [vk@ufal.mff.cuni.cz](mailto:vk@ufal.mff.cuni.cz)

Abstract:

Interactive Question Answering (IQA) is a natural and cohesive way for a user to obtain information by interactive with a system using natural language. With the advancement in Natural Language Processing, research in the field of IQA has started to focus on the role of semantics and the discourse structure in these systems. The need for a deeper analysis, which examines the syntax and semantics of the questions and the answers is evident. Using this deeper analysis allows us to model the context of the interaction. I will look at a current closed-domain IQA system which is based on Linear Regression modeling. This system uses superficial and non-semantically motivated features. I propose adding deep analysis and semantic features in order to improve the system and show the need for such analysis. Particular attention will be placed on the so-called follow-up questions (questions that the user poses after having received some answer from the system) and the role of context. I propose that adding the linguistically heavy features will prove beneficial, thereby showing the need for such analysis in IQA systems.

Keywords: interactive question answering, context modeling

# List of Tables

4.1	Mean rank of correct A2 out of 306 answer candidates in single feature main effects only models. . . . .	53
4.2	Mean ranks of correct A2 of out 306 answer candidates in interaction models. . . . .	55
4.3	The predictors for model 4.6. . . . .	56
4.4	Number of positive instances extracted for Q1-Q2 context defining features (out of 1522 possible ones). . . . .	59
4.5	Number of positive instances extracted for A2-finding features (out of 465732 possible ones). . . . .	59
4.6	Number of instances extracted for each possible Q1-Q2 context defining transition features (out of 1522 possible ones). . . . .	59

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	The History . . . . .	7
1.1.1	TREC . . . . .	7
1.1.2	CLEF . . . . .	8
1.2	Multidisciplinary and Cross-Field Aspects of IQA . . . . .	9
1.2.1	Multidisciplinary Aspects of IQA . . . . .	9
1.2.2	Cross-Field Aspects of IQA . . . . .	10
1.3	Classical Architecture of IQA Systems . . . . .	11
1.3.1	Question Analysis . . . . .	11
1.3.2	Document Retrieval . . . . .	12
1.3.3	Answer Extraction . . . . .	12
1.3.4	Answer Evaluation . . . . .	13
<b>2</b>	<b>The Problem: Modeling Context</b>	<b>14</b>
2.1	Interaction as a Form of Dialogue . . . . .	14
2.2	Context in IQA . . . . .	16
2.3	Follow-Up Questions . . . . .	17
2.3.1	Anaphora . . . . .	18
2.4	Modeling Context: Chai and Jin . . . . .	19
2.4.1	Discourse Roles . . . . .	19
2.4.2	Discourse Transitions . . . . .	21
2.5	Modeling Context: Sun and Chai . . . . .	23
2.5.1	Centering Theory . . . . .	24
2.5.2	Three Proposed Models . . . . .	26
<b>3</b>	<b>The Solution: Context Modeling Using Deep Analysis in Bob</b>	<b>29</b>
3.1	Bob: A Virtual Librarian . . . . .	29
3.2	Logistic Regression . . . . .	30
3.3	Context Features . . . . .	33
3.4	Chai and Jin . . . . .	33
3.4.1	Constraint Refinement . . . . .	39
3.4.2	Participant Shift . . . . .	40
3.4.3	Topic Exploration . . . . .	41
3.4.4	Topic Activity shifts to Topic Activity . . . . .	42
3.4.5	Topic Activity shifts to Topic Entity . . . . .	43
3.5	Sun and Chai . . . . .	44
3.5.1	Reference Model . . . . .	47
3.5.2	Forward Model . . . . .	48

3.5.3	Transition Model . . . . .	48
<b>4</b>	<b>Evaluation and Discussion</b>	<b>51</b>
4.1	Evaluation . . . . .	51
4.1.1	Data . . . . .	51
4.2	Results . . . . .	52
4.3	Discussion . . . . .	57
4.3.1	Issues to Consider . . . . .	58
<b>5</b>	<b>Conclusion</b>	<b>61</b>
	<b>References</b>	<b>62</b>
	<b>Appendices</b>	
<b>A</b>	<b>Sample Bob Snippet Data</b>	<b>65</b>
<b>B</b>	<b>Stanford Parser Dependency Relations</b>	<b>69</b>
<b>C</b>	<b>MiniProbBank Sample</b>	<b>71</b>
<b>D</b>	<b>GuiTAR Output Sample</b>	<b>72</b>

# Chapter 1

## Introduction

Interaction is one of the main forms of communication amongst humans. One of the main goals of communication is the exchange of information. In today's world, with the advances in computer science and other technologies, the amount of available information has become vast. The problem, therefore, has become finding a way to access this information.

In computer science the field that deals with the accessing of information is called Information Extraction (IR). In IR most often queries are used to obtain information from a database or a collection of documents. Queries often require some type of previously learned knowledge of the query language. Using natural language to access a database or get information is therefore often more useful. This is exactly what a question answering system does. That is, a question answering system looks for ways of obtaining answers and information from a system using natural language.

Classical question answering systems aim at finding the relevant answer for a question presented in natural language. These systems look at each question individually and the answer from one question is never taken into account for a following question. In a sense, this limits the role of the user, as well as that of the answer. First, the role and desire of the user is very important, since the goal of any such system is to successfully provide the user with the desired information. Second, it is often upon hearing an answer that we further think of information that we wish to obtain and questions that we may wish to pose. In natural dialogue, we use the information gained from the conversation and the previous exchanges in later exchanges. Taking these issues into consideration, the field of *Interactive Question Answering* (IQA) has been born.



In IQA, the user and system takes turns asking questions and providing answers. Not all IQA systems allow questions which follow-up on something the system has previously answered or that the user has previously asked. However, in all these systems the user dynamically interacts with the system by posing questions and receiving answers, all in natural language.

IQA systems are therefore much more natural, since they follow the structure of natural dialogue where previously mentioned information is taken into account. This allows for follow-up questions, which are especially beneficial when the user wants to clarify something, obtain more information on a given topic, or may simply want to reformulate the question. Further, these systems give a greater role to the user, since he/she is able to interact with the system in order to get the precise information that he/she is looking for.

## 1.1 The History

The history of IQA dates back to the 1970's. One of the first IQA systems is LUNAR. LUNAR was originally developed for NASA as a system which enabled a geologist to evaluate and compare data on lunar rock and soil composition. The system dealt with a limited domain, and required interaction with the user. Another example is TRYPSIS, an IQA system which used continuous speech understanding in order to help users gain information about making trips. [27] While these systems were not very advanced, they demonstrate the essence of IQA systems: the need for *interaction* between the user and system, as well as a way for the user to *gain information* from the system.

### 1.1.1 TREC

The Text Retrieval Conference (TREC) was started in 1992 as a way to support text retrieval research. Since then, TREC has become one of the leading conferences for QA systems. [24] One of the most important things about TREC is that it provides standardized data on which different QA systems can be evaluated on. Moreover, this standardized data can be used to compare systems amongst each other and to be able to set benchmarks. This not only promotes competition and research, but also allows people to track the development and improvement of systems over time.

Given the growing interest in IQA, TREC in 2006 introduced the complex interactive question answering task (ciQA). [12] As the name indicates, ciQA looks at complex questions (i.e. not fact-based questions) which include some form of interaction with the user. A human assessor is used to evaluate the performance

of each system. This new recent TREC extension shows how interaction in QA systems is becoming more and more popular. However, the development of IQA at TREC is complex. First, there is the problem of standardized data and results. For non-interactive QA, a precise definition of a correct answer exists. However, this is not the case for IQA. It is easy to define a precise answer for a factoid question: it consists of a single word or expression. However, in IQA, most often a correct answer cannot be represented as a single word and/or expression, since the user might want something more than a simple fact. Second, with interaction there is information that is carried on during the interaction session. That is, the question a user might pose to the system will most probably depend on what the user has previously asked or the information he/she has obtained. However, it is hard to compare systems in this sense, since the previous questions and answers will not be the same. As noted above, the ciQA part of TREC requires a human annotator to evaluate the systems performance. Due to these reasons, the development of IQA in TREC has not been as prompt as for non-Interactive QA.

### 1.1.2 CLEF

Similarly to TREC, there is the Cross Language Evaluation Forum (CLEF), which also promotes research and development in the field of information retrieval. The additional thing about CLEF is that it promotes monolingual as well as multilingual research in the field, concentrating mainly on European languages. Like TREC, CLEF provides standardized data that can be used to test systems and to compare them against each other. CLEF introduced an interactive track in 2001, named iCLEF. [4] iCLEF does not only deal with IQA, but also with interactive information retrieval (for example text-based image retrieval).

The tasks in iCLEF (and CLEF) are different from those in TREC since they focus on multi-lingual tasks. [18] For example, in iCLEF 2004 the task involving IQA required that the language that the question was posed in be different from the language that the document collection is in. While this multi-lingual aspect makes the tasks in iCLEF different from those in TREC, both these conferences show how the role of interaction is becoming more and more relevant in question answering and information retrieval research.

## 1.2 Multidisciplinary and Cross-Field Aspects of IQA

### 1.2.1 Multidisciplinary Aspects of IQA

IQA is a multidisciplinary field that encompasses research from a wide variety of research areas. This multidisciplinary aspect reflects the complexity of the task, since it required cooperation and working together of people from different fields and backgrounds.

#### **Computer Science**

Perhaps most obviously, IQA is highly connected to the field of computer science. The system that the user interacts with is a computational system. Information storage, information retrieval, databases and search are just some of the issues in computer science that IQA touched upon.

#### **Linguistics**

One of the requirements of any IQA system is that the questions and answers must be in natural language. It is therefore clear that any such system must involve the study of natural language, or linguistics. Different areas of linguistics, such as syntax, semantics and phonetics have all been useful in IQA research. That is, these different levels of analysis allow people to explore and understand different levels of linguistic meaning. The development of the field of computational linguistics (an interdisciplinary branch of computer science and linguistics) has been especially important for IQA.

#### **Psychology**

Thirdly, there is the role of psychology. In studying human-human dialogues, one must look at issues such as the expectations of each of the two (or more) speakers, as well as the style of interaction (whether it be speech or text). In such interactions, both the speakers have expectations about what it is they wish to gain from this conversation and well as a mental model of what it is the other person want and expects. In IQA the situation is different since the communication occurs between a human and a machine. Therefore, in designing such a system one must look at this human-machine interaction in order to examine and study the role and expectations of the user. In examining these expectations one will be able to better understand the user, and therefore will be able to design systems which better model the users' wishes. These

considerations are also important in designing the interface and decided the mode of interaction that the system will encompass.

### **1.2.2 Cross-Field Aspects of IQA**

IQA, more than just being multidisciplinary, draws on several research fields. The three fields that it draws most from are Natural Language Processing, Information Retrieval and Human Computer Interaction.

#### **Natural Language Processing**

Natural Language Processing (NLP) looks at the interaction between computer science and natural (human) language. Since IQA systems require that the questions and answers both be in natural language, it is quite obvious why this field is of great importance. The system must be able to in some way *understand* the question provided in natural language in order to look for an answer. Further, it must be able to synthesize natural language in order to be able to provide the user with an answer in natural language. There are many different methods that have been developed in the last several years. Tagging, chunking and parsing are just a few which can be very useful in analyzing the questions and answers. These methods allow for the analysis of the structure of the sentences being dealt with, as well as the ability to interpret and deal with the information that the text contains.

#### **Information Retrieval**

Information Retrieval (IR) is a branch of computer science that provides methods for searching through documents and data. This search can involve different types of data, as well as document collections. In the past several years, as computer memory has become cheaper, the amount of data which can be stored has been constantly increasing. However, a distinction must be made between the available and relevant data. IR provides techniques for retrieving relevant information in an efficient manner. IQA relies heavily on this field, since the system must find the answer within some sort of data collection that it has access to. Finding and relating only the necessary information back to the user is the key to a successful and precise IQA system.

#### **Human Computer Interaction**

The word interaction in IQA is very important. That is, one must not forget the role of the interaction between the user and the system. The field of Human Computer Interaction (HCI) studies this interaction. Research in this field is

important so that the requirements of the user are reflected and taken into account by the system. That is, the system needs to represent what the user expects and knows. The system needs to acknowledge in some fashion the needs of the user in order to foster and stimulate interaction.

The role of HCI is also important in designing the interface for the IQA system, for this is the basis of the interaction between the system and the user. In designing an IQA system, the developer must consider what he/she wants the user interface to appear like. The choice of designs depends on many things such as the environment that the interaction will take place in (i.e. should it be available on the web), the type of users (i.e. are they experts in the field or laypeople) and the type of information provided (i.e. are the answers represented as text, images, or video).

### 1.3 Classical Architecture of IQA Systems

Many different IQA systems exist that deal with different domains and are implemented using different techniques. Some are based on machine learning approaches and some rely more on work from computational linguistics. Some use logic to reason about the underlying meaning of the question while others look for keywords found in the question. While there are many variations, there is also a commonality to all these systems. Below I describe the common parts of the IQA system architecture that are found in most (if not all) systems. The exact implementation of each of the parts depends on the particular system and its goals.

#### 1.3.1 Question Analysis

Every system must first start by analyzing the question that the user provides. This question is provided in natural language and the system then converts this question into a form that it can use to reason and obtain an answer with. Different systems do this differently. For example they might convert it to a query, or might extract keywords or look for some form of pattern matching. This process can also be thought of as *question understanding*, meaning that the system is trying to *understand* the question in order to be able to provide a relevant answer.

Question analysis can be done on many different levels depending on the level of processing, from the shallow (i.e. doing regular expression pattern matching) to the deeper (i.e. involving syntactic and semantic processing). As mentioned above, one can simply do pattern matching or look for extracted keywords. But

language is a complex phenomenon. There are different words which can be used to express the same thing. Also is this enough? Most often IQA systems do some kind of syntactic and semantic interpretation an analysis. This deeper analysis allows for a richer understanding of the language and the meaning being conveyed by the language.

### **1.3.2 Document Retrieval**

Once the system has analyzed the question and has extracted the necessary information and/or has converted the question into an internal representation, the system then searches for a suitable answer. Where the system searches for the data depends on the implementation. It might have a pre-selected set of answers or documents, or it might use the web. The type of data you would use depends greatly on the domain and scope of the system. The more open in terms of domain the system is, the larger this data will have to be.

Regarding the size of the data, there is the concept of speed which must be considered by IQA systems. That is, a user does not want to have to wait a long time to receive an answer to a posed question. Any such system, therefore, must return answers within some reasonable time limit. This means that the processing time needs to be monitored. This processing time does not only depend on the size of the data collection, and this issue should be kept in mind during the development of each stage of the system.

### **1.3.3 Answer Extraction**

Once the document or data that contains the answer is found, the precise answer must be extracted (if the answer is not already in this precise form). That is, in some cases during the search and document processing phase, an IQA system will return a document or a passage that contains more information than the question asked for. Therefore, the system must extract only the relevant information. Also, in this stage a system might return several possible answers. Some systems provide several possible answers, although this is rare. Human dialogue usually does not involve multiple possibilities for each utterance. Therefore, most systems will rank the possible answers and return the one with the highest ranking. There are many different proposed algorithms as to how to proceed with the ranking. This is especially important in instances where the system finds the correct answers, but does not rank it as the most correct one. Clearly, the answer must be in natural language.

### 1.3.4 Answer Evaluation

The notion and role of evaluation is important for any scientific field of study. First, and perhaps most obviously evaluation is important since it provides us with a way of quantitatively examining the performance of a given system or algorithm. Further, evaluations are used to compare the performance of different systems on the same task. The evaluations must be done in a controlled and carefully set-up fashion. That is, the evaluation and evaluation metrics used must be valid and the settings of the experiments must be set up in a way to avoid bias and outside influences. This is important not only for IQA, but any experimental design.

In IQA, the question of evaluation is also important because there is no one way to do evaluate such a system. As mentioned above, there is no one standard definition of an answer in IQA. Also, systems are designed differently depending on what task the user is focusing on. Therefore, for a task that involves information gathering a much more precise answer is expected than for a task which involved information browsing. In a sense, the evaluation of the answer depends on the user and whether he/she is satisfied with the obtained answer. It might be the case that the user has obtained the desired information and he/she will end the interaction at this point. It might be the case that the answer will prompt the user to ask further questions, and the interaction will continue. Or, the answer provided will not be satisfactory to the user, and he/she will attempt to clarify or reformulate the question. The precise evaluation used for any given IQA system will, therefore, have to be designed by taking into account both the task at hand and what constitutes a correct answer for a given user. [20]

## Chapter 2

# The Problem: Modeling Context

The main difference between a QA system and an IQA one is the notion of *interaction*. IQA systems allow for communication between the user and the system and this communication does not take place in a vacuum. Rather, it occurs in some context. What this implies is that for IQA the context that the interaction takes place is crucial. In other words, one must find a way of modeling context.

Modeling or representing context is not an easy task. It requires for the knowledge from the dialogue to be represented, and used (if necessary) in interpreting a given utterance.

### 2.1 Interaction as a Form of Dialogue

In an IQA system, the exchanges between the user and the system can be thought of as a form of dialogue. The user and the system take turn asking and answering questions, respectively. [26] There has been a lot of work done in linguistics about dialogues and the different ways to model them.

In his book *The Interactive Stance* Ginzburg examines how meaning is brought about in conversations, or dialogues. [9] In his analysis of dialogues, he lists three dominant features which are always present and necessary: coherence, conciseness and radical context dependence.

- (1) (a) Emma: We have a flat.



- (b) Robert: Ah, I see. (Pause) Nice? (Pause) A flat. It's quite well established then, your...uh...affair?
- (c) Emma: Yes.
- (d) Robert: How long?
- (e) Emma: Some time.
- (f) Robert: But how long exactly?
- (g) Emma: Five years.
- (h) Robert: Five years?

First, dialogues are coherent in the sense that a given utterance will follow, both in logic and meaning, the preceding utterance(s). In other words, there is a natural flow from utterance to utterance, which makes the conversation logically consistent. An answer will be related to a previous question, such as, for example, the confirmation *yes* in (c) will confirm what was previously said in 1(b). This confirmation is done in such a way that the utterance in 1(c) logically follows the utterance in 1(b).

When comparing dialogue to regular text, most often a dialogue is much more compact. In a sense, a dialogue is much more efficient. In the given dialogue excerpt above, the interaction is very brief - it contains only thirty one words. However, the amount of information contained is not small. We know that Emma has a flat, that she has a well-established affair that has been going on for five years. We also know that Robert did not know about this flat, nor did he know the length of the well-established affair. Further, we also know that Robert is surprised by the length of the affair. Portraying all this information in a text format would require more than thirty one words.

Finally, as Ginzburg finds, there is the dependence of the dialogue on context. Many utterances in a given dialogue, if taken in isolation will lose their meaning. In other words, many utterances cannot stand on their own and require the knowledge of previous information in order for them to be properly assessed and understood. For example, the *yes* in 1(c) or *how long* in 1(d) cannot be properly interpreted and understood if the previous section of the dialogue is unavailable. Ginzburg goes further to state that this dependence of context is *radical*, placing an emphasis on this context and its necessity in correctly interpreting and understanding dialogues.

In looking at the example above, and in examining sentences such as 1(c) and 1(d), how exactly does one understand and correctly interpret what is being said? There seem to be two main components. The first is the semantics of the

word(s) of the given utterance. In sentence 1(c), we know that the *yes* stands for affirmation. This can be thought of as the meaning of this word that can be found in a dictionary. However, there is also the context that this word occurs in. For example, in this particular case the *yes* refers to the speaker's (Emma's) well established affair. That is, Emma is confirming to Robert that the affair is well-established. Changing one of these two components changes the interpretation of this utterance. For example, if instead of *yes* in 1(c) we had *no*, then the meaning would be different. That is, in this case Emma would be letting Robert know that the affair is not well-established. Likewise, if the utterance was *yes* but the previous utterance asked for a confirmation about Emma's age, the interpretation would be different as well.

Ginzburg addresses this issue as well. He states that given an utterance, it's *content* is attained by combining the *meaning* of the word(s) and the *context* that they occur in. In a sense, the meaning of given utterance, or expression exists outside the notion of space and time. The meaning of the utterance in 1(c) or 1(d) will be the same no matter what dialogue, text, or conversation they occur in. However, to understand the full content of such an utterance it must be placed in its appropriate context - i.e. it must have a time and place associated with it.

## 2.2 Context in IQA

Dialogue theory shows us that context and the modeling of context is crucial in fully understanding dialogue. This can be extended to IQA. Since IQA can be regarded as a dialogue between the user and the system, to get at the full sense of such a dialogue one must not only consider the semantics and meaning of the utterances but the context of the dialogue as well.

In a dialogue between two humans, the context is in some sense inherent. The participants possesses the knowledge of space and time, declarative information about the world (such as the fact that World War I began in 1914), his/her beliefs about the world and the information that has already been mentioned in the interaction. Modeling discourse or context in the case of IQA systems is difficult because it requires for the computational systems to be able to have access to the information and knowledge that for humans is inherent.

## 2.3 Follow-Up Questions

As previously stated, in any interaction, whether it is between two humans or between a human and a computation system, utterances (very often) cannot be understood in isolation. The context of the utterance can involve many things, such as the time and location of the interaction, the beliefs of the user(s), etc. However, there is also the context that is created by the interaction itself. That is, one must know and understand what has previously been said in the interaction in order to understand the current state.

- (2) (a) Can I take out a book?
- (b) If the item is available in the Library you can check it out immediately. If you order a book from Brixen-Bressanone for Bozen-Bolzano or the other way round it normally takes two days to arrive. If you choose Bruneck-Brunico or Schlanders-Silandro as destination will take a bit longer. We will let you know as soon as it's there.
- (c) What about a DVD?
- (d) In Brixen-Bressanone the DVD collection is on the first floor. In Bozen-Bolzano you have to order them via the OPAC and pick them up at the information desk. DVD can be kept for two weeks and the loan cannot be extended.

In examining the dialogue excerpt in 2, we can understand 2(a) in isolation since this is the beginning of the interaction and this is the first utterance. 2(b) is an answer by the system to the user and is of course related to the posed question. In order to understand 2(c), however, one must have access to the information contained in utterance 2(a).

- (3) Can I take out a DVD?
- (b) If you want to borrow books, DVDs or journals you need to take them from the shelf yourself and check them out at the information desk. If you want them to be sent to Brixen-Bressanone, Bozen-Bolzano, Bruneck-Brunico or Schlanders-Silandro you can order them via the OPAC.
- (c) What is OPAC?
- (d) What I call the OPAC is the online public access catalogue of the Library of the Free University Bozen-Bolzano. With the OPAC you can search all items kept at the Library.

And you can also borrow or reserve them.

Looking at the dialogue excerpt in 3, 3(a) can be understood in isolation and 3(b) is the answer to 3(a). However, in order to understand 3(c) one must have access to the information in 3(b). What we can see in these two excerpts is that in order to understand a subsequent question(s), one might need information previously mentioned in the interaction. In the case of 2, the question is related to the previous question. In 3, the question is related to the previous answer. The questions in 2(c) and 3(c) are called follow-up questions, since they follow-up, or expand on some previously mentioned information.

Of course, a question does not have to be related to a previous utterance (whether it be a question or answer), as can be seen from example 4.

- (4) (a) What are the opening hours?
- (b) You can find our opening hours on this website.
- (c) What kind of databases do you have?

However, most often in dialogue, whether it is human-human or human-computation system, there is a coherence between the utterance(s), which implies that it builds on what has been previously stated or said. Moreover, while the frequency of related follow-up questions may vary from interaction to interaction, this is an important issue which must be addressed by IQA systems [5, 2, 25].

### 2.3.1 Anaphora

The excerpts in examples 2 and 3 above show how follow-up questions can be related to previous utterances, whether they are previous questions or answers. This relatedness is in terms of the meaning of the utterances. The excerpt in 5 is another such example.

- (5) (a) Can I borrow a book?
- (b) Books can be kept for four weeks. And you can extend the loan period twice - again for four weeks. But this is only possible if the book is not already reserved by someone else.
- (c) Can I reserve it?

However, while example 5 also includes a follow-up question like 2 and 3, it differs from the other two examples in that the follow-up question, while asking

about reserving books, does not explicitly mention the word *books*, but rather uses the pronoun *it*. In natural language, whether it is dialogue or text, humans use pronouns in order to refer to a noun which has previously been mentioned. As humans, we inherently and automatically know what such a pronoun is referring to (except in the rare ambiguous cases).

Phenomena portrayed in examples such as 5 are referred to as anaphora. Anaphora is a linguistic phenomenon where the meaning of a word or expression is given by its reference to another word or expression, called the antecedent. Anaphoric resolution is the process of finding the antecedent to which the pronoun is bound to (i.e. the word that the pronoun refers to). In IQA, to fully understand any given follow-up utterance the system must be able to resolve anaphora. [21] In 5(c) the antecedent of *it* is *books* in 5(b). Antecedents need not be found in immediately preceding utterances. However, most often the antecedent can be found very close to the pronoun. [2] Once again, while this is a simple task for humans it is a rather complex one for computational systems. This is especially true in cases where there are multiple candidates for the antecedent.

Given the importance of context in IQA, many theories and approaches have been proposed for dealing with this issue.

## 2.4 Modeling Context: Chai and Jin

In looking at IQA in their paper *Discourse Structure for Context Question Answering*, Chai and Jin identify and stress the need for modeling context in IQA. They propose a semantically-rich discourse representation which is based on discourse roles and discourse transitions [chai-jin].

### 2.4.1 Discourse Roles

In an IQA dialogue, each utterance carries meaning about the syntactic entities that it represents such as noun phrases, verb entities, etc.

- (6) (a) What is OPAC?
- (b) Can I search for a book using OPAC?

For example, in 6, 6(a) is a question that concerns the noun entity *OPAC*, while 6(b) concerns itself with the task of *searching* for a *book*, using *OPAC*. The type of semantic information can be obtained by doing syntactic analysis of the utterances in question. However, each utterance has a semantic role as related

to the entire dialogue, or discourse. This is what Chai and Jin call discourse roles.

What these discourse roles need to represent is the information and intentional aspect of each utterance. Information aspect refers to the semantic information that is carried by the syntax and the semantics of the utterance. The intentional aspect refers to the intentions of the user with the respect to the system and to the system as a whole. Chai and Jin identify three main discourse roles: intent, content and media.

### **Intent**

Intent refers to the intentional perspective of the given utterance. In other words, this is the purpose of the question and of the user.

- (7) (a) For how long can I keep books?
- (b) Loan periods depend on the type of medium.  
Do you have a normal book? Or is it something from the textbook collection? Or maybe a journal, a DVD or something ordered from another library via Interlibrary Loan?
- (c) A normal book.

Example 7(a) is an example of an information request (the user is asking for how long he/she can keep a book that has been borrowed from the library). However, 7(b) is a confirmation type of question, where the user is confirming something that has been asked by the system. Therefore, 7(a) and 7(b) have different purposes.

Intent can be defined using two features: act and motivator. Act refers to whether the user is asking the system for information or whether he/she is replying to a system question. Motivator describes the goal of the user; whether he/she is trying to retrieve or confirm information. Therefore, for 7(a) the act is a request for information and the motivator is an answer request. For 7(b) the act is a reply and the motivator is also an answer request.

### **Content**

Content refers to the informational perspective of the given utterance. This is the semantic information that is being brought to the discourse by the utterance. Content can further be subdivided into target, topic and focus. Target is related to the expected answer type. Expected answer types include entities

and propositions.

- (8) (a) When is the library open?
- (b) How do I borrow a book?
- (c) How do I borrow a DVD?

For example, for the given question in 8(a), the expected answer type would require a noun entity of the type *time*, since the question is a when question. Example 8(b) has a propositional expected answer type, since it is a *how* questions that requires the answer to give a procedural account of *how* to do the stated activity (in this case borrowing a book). Expected answer types are useful since they can help narrow down the set of suitable answers for the given question.

The topic is the scope of the given utterance. Focus is a narrowing of the topic, and can be thought of as the current focus of attention on the given topic. Looking at example 8, we can see that 8(a) and 8(b) have a different topic: 8(a) is concerned with the opening hours of the library (i.e. the entity *library*), while 8(b) is concerned with the activity of borrowing something from a library (i.e. the activity of *borrowing*). The focus of 8(a) is the opening hours of the library, while the focus of 8(b) is books.

In looking at 8(b) and 8(c) however, we can see that they have the same topic but a different focus. That is, they are both concerned with the activity of *borrowing*. However, the focus of 8(b) is *books*, while the focus of 8(c) is *DVDs*.

### Media

In IQA, the user, by asking a question is trying to obtain some type of information. The system's role, therefore, is to present the information that the user is seeking. This information can be presented in different ways, or media. Media can be further subdivided into format and genre. Format refers to the style of the output: is it text, an image, speech segment, etc. Genre on the other hand specifies the needs of the answer. In other words, the genre specifies what type of information that the answer needs to provide. For example, different genres include fact, lists, summaries, etc.

### 2.4.2 Discourse Transitions

The discourse roles more or less look at utterances in isolation. That is, each utterance that is a question will have its intent, content and media defined.

However, in an IQA dialogue or discourse the transitions from question to question carry information as well. Therefore, Chai and Jin propose looking at and identifying these transitions according to the relation of a question and the question before it. That is, transitions are identified for pairs of questions (a question and the follow up question, with the answers not being taken into consideration). They identify three types of transitions: topic extension, topic exploration and topic shift.

Topic extension can be classified as a question which talks about the same topic as its previous question, but with different participants or constraints. This type of transition has two subcategories: constraint refinement and participant shift. Constraint refinement represents the case when the two questions have the same topic but the second question has additional constraints that are not present in the first question. Participant shift represents the case when the two questions have the same topic, but with different participants.

- (9) (a) Can I borrow books?  
(b) Can I borrow books in Brixen?  
  
(c) Can I borrow books?  
(d) Can I borrow DVDs?

Looking at example 9, 9(a) and 9(b) are an example of constraint refinement, and 9(c) and 9(d) are an example of participant shift.

### **Topic Exploration**

Topic exploration can be thought of as two consecutive questions which have the same topic, but a different focus. In a sense, the second question is exploring some other aspect of a given topic. Therefore, for this transition there is similarity to the previous question, but less so than in the case of topic extension.

- (10) (a) Where can I borrow books?  
(b) How can I borrow them?

Example 10 shows an example of topic exploration. Both questions are talking about the activity of *borrowing books*, while the 10(a) is asking *where* one can do this, and 10(b) is asking *how* one can do this.



## Topic Shift

Topic shift refers to the case where the two consecutive questions are concerned with two different topics. There are two subtypes of topic shift: activity topic shifts to another activity topic and activity topic shifts to entity topic. Activity topic shifts to another activity topic is the case where the topic shift from the first to the second question and both the questions concern themselves with activities. The activity in the first question is related (but not the same) as the activity of the second question. Activity topic shifts to an entity topic is the case where the first question concerns itself with an activity and the second with an entity, both having different topics. However, the entity of the second question is related to the activity of the first.

- (11) (a) Can I borrow books?
- (b) How can I reserve books?
  
- (c) Can I borrow books in Brixen?
- (d) Where is the library in Brixen?

In example 11, 11(a) and 11(b) show an example of an activity topic shifts to another activity topic. That is, the topic of 11(a) is borrowing, and the topic of 11(b) is reserving, while both 11(a) and 11(b) examine activities. 11(c) and 11(d) show an example of an activity topic shift to an entity topic. The topic of 11(c) is borrowing books (an activity) and 11(d) is the library in Brixen (an entity).

While the examples above, and topic shift in general show a shift in topic the two questions are still related since they do not talk about two completely unrelated topics.

Chai and Jin propose that in an IQA setting, context can be modeled using the above specified discourse roles and discourse transitions. The extraction and analysis of these roles and transitions requires deep syntactic and semantic processing and is therefore considered a semantically-rich approach.

## 2.5 Modeling Context: Sun and Chai

Another approach to modeling context has been proposed by Sun and Chai in their paper *Discourse Processing for Context Question Answering Based on Linguistic Knowledge*. [23] Sun and Chai present another way of modeling context

as a way of better understanding IQA dialogues in order to facilitate answer retrieval. The model, like the Chai and Jin model, looks at how consecutive questions are linked and how the way they are linked can be beneficial. To look at the different manners in which consecutive questions can be linked, they use Centering Theory.

### 2.5.1 Centering Theory

Centering Theory is a theory that has come out of linguistic research, which examines how discourse can be modeled by looking at the entities or centers that it mentions. That is, centers can be thought of as semantic entities which are a part of the discourse model. Intuitively, we can think of centers as the entities which have the key roles in each sentence. In a given discourse, each utterance has its own set of semantic entities or centers [10, 19].

- (12) (a) When is the library open?
- (b) Where is the library in Bolzano?

For example, in utterance 12(a) the center is *library*, while in utterance 12(b) there are two centers, namely *library* and *Bolzano*.

The idea behind Centering Theory in discourse is that utterances which have the same center(s) will have something in common. In a sense, utterances which are in part or whole talking about the same entities will have the same centers.

- (13) (a) Can I take out books?
- (b) Can I take out books in Brixen?
- (c) Where is the library in Brixen?

#### Forward Looking Centers

Forward looking centers, or  $C_f(U_n)$ , are defined as the set of entities which are mentioned in utterance  $U_n$ . Intuitively, we can think of these entities which might be mentioned in future utterances. That is, these are the entities that the future dialogue might mention.

Therefore, looking at example 13, the forward looking centers for 13(a), 13(b) and 13(c) are *books*, *books*, *Brixen* and *library*, *Brixen*, respectively.

### Backward Looking Centers

Backward looking centers, or  $C_b(U_{n+1})$  is the most highly ranked forward looking center of  $U_n$  that is mentioned in  $U_{n+1}$ . Intuitively, we can think of this as the entity in  $U_n$  which can be found in  $U_{n+1}$ , and in a sense links these two utterances together.

Looking at example 13, there is no backward looking center for 13(a), since it has no previous utterance. For 13(b) the backward looking center is *books*, while for 13(c) it is *Brixen*.

### Preferred Centers

The preferred center is the highest ranked member of the forward-looking centers. This can be thought of as the entity that is the most relevant for the given utterance.

Looking at example 13, the preferred centers for 13(a), 13(b) and 13(c) are *books*, *books*, and *library*, respectively.

### Transitions

Further, Centering Theory defines three different types of transitions between two consecutive utterances: continuation, retaining and shifting. The different transitions show how topics change or remain the same as the discourse progresses. In some sense, the entities or centers of a given utterance can be thought of as its topic(s).

The transition between two utterances, namely  $U_n$  and  $U_{n+1}$  are based on two conditions:

- (i) The backward looking centers of the two utterances are the same:

$$C_b(U_{n+1}) = C_b(U_n)$$

- (ii) Whether the backward looking center of  $U_{n+1}$  is the most preferred forward looking center of the same utterance:

$$C_b(U_{n+1}) = C_p(U_{n+1})$$

Using these two criteria we can define the three transitions. Continue is the case when both conditions hold. Retain is defined as the case when (i) holds but (ii) does not. Finally, shift is defined as the case where (i) does not hold.

### 2.5.2 Three Proposed Models

Using Centering Theory, Sun and Chai propose three different models which can be used for modeling context. The basic notion in Centering Theory are of course the centers themselves. Moreover, the centers are based on the entities, which might not be the words themselves which are present in the utterance. Looking at example 14, we can see that the forward center of 14(a) is *DVDs*. Utterance 14(b) also talks about DVDs, but it mentions them by using a pronoun. However, the forward center for 14(b) is *DVDs* as well, even though this word is not explicitly mentioned. Therefore, in order to use these models in practice and in dialogues we must resolve pronouns, since in the case where we encounter a pronoun the entity for the given utterance is the entity that the pronoun is referring to. In other words, in order to develop a working model and an algorithm which uses Centering theory and the notion of centers, we need to be able to resolve anaphora.

- (14) (a) Can I borrow DVDs?  
(b) How can I search for them?

#### Reference Model

This is the most basic model. The idea is that by resolving pronouns, we can use these antecedents to help guide our search for the correct answer. Therefore, in this model for a pronoun in a given question the antecedent is found and added to the query of the current question. Looking at example 14(b) above, using this model we would resolve *them* to *DVDs*, and would add the term *DVDs* in formulating the query for question 14(b).

#### Forward Model

This model goes one step further than the reference model. In this model, for a given question (or utterance), we add the forward looking centers of the previous question to the query of the current one. This model inherently contains the reference model, and it includes anaphoric resolution as well. This model is based on the idea that in discourse there is a coherence, and that one does not jump from topic to topic as the discourse progresses. Rather, it is more likely that subsequent questions will in some way be related to what has already been mentioned.

- (15) (a) Can I search for books using OPAC?  
(b) Where can I access it?

In example 15, the forward centers of 15(a), namely *books*, *OPAC* would be added to the query being made for 15(b).

### Transition Model

The transition model goes one step further than the forward model. That is, it looks at the specific relation between the centers of two consecutive questions. It does so by examining the different types of transitions which can occur between utterances, as defined by Centering Theory. This model, inherently includes the forward and reference models. It also therefore needs to include anaphora resolution. The transitions are defined, as above, using Centering Theory. In this model, the *shift* transition is further subcategorized into *smooth shift* and *rough shift*, to show the different types of shifts. The model proposes an algorithm for detecting the different types of transition.

Given the NP which defines the preferred center, the NP has a modifier and a head.

NP Modifier	NP Head	Transition
Same	Same	Continue
Different	Same	Retain
Same	Different	Smooth Shift
Different	Different	Rough Shift

- (16) (a) Where is the Brixen library?  
 (b) When is the Brixen library open?
- (c) How do I search the book collection?  
 (d) What about the DVD collection?
- (e) How do I perform a book search?  
 (f) Where can I find the book stacks?
- (g) How can I search the book collection?  
 (h) When is the Brixen library open?

Looking at the definitions of the four transitions, we can see that they represent the different ways in which the two consecutive utterances are linked, with respect to their centers. Continuation can be defined as the case when the two consecutive utterances are talking about the same entity. The utterances

16(a) and 16(b) show an example of such a transition, since they are both talking about the *Brixen library*. Retain can be thought of as the transition where the two utterances are talking about the same entity but with a different constraint. Example 16(c) and 16(d) show an example of such a transition. Both the utterances are talking about *collections*, but 16(c) is talking about *book collections* and 16(d) about *DVD collections*. Smooth shift is the case where the entities being mentioned are different for the two utterances, but there is some similarity between the two, though small, represented by the same modifier. Utterances 16(e) and 16(f) show such an example. They talk about different entities, namely *search* in 16(e) and *collections* in 16(f). However, they are both asking about entities related to *books*. Finally, rough shift is the case when there is no relation between the entities of the two given utterances. Example 16(g) and 16(h) show such an example, the first having *book collection* as its center, and the second *Brixen library*.

The two models proposed by Chai et Jin and Sun et Chai both show ways that context can be modeled. That is, they both show a way of representing context in a discourse structure, and a way of modeling how information is carried on within such a dialogue.

## Chapter 3

# The Solution: Context Modeling Using Deep Analysis in Bob

### 3.1 Bob: A Virtual Librarian

Bob is an IQA system that has been developed for the library at the Free University of Bolzano-Bozen. The system is web-based (i.e. available on the university website), is bi-lingual (allowing interaction in English and German), and can handle questions in the university library domain. In other words, Bob is a closed-domain IQA system [3, 13].

Bob was first created using a bottom-up approach. First, a knowledge base was created representing the set of possible answers. The knowledge base was created especially for this system. Since it involves library-specific info, data was collected using several resources. First, past queries found in archived emails were used to extract possible questions and their related answers. Second, information was extracted from the library's frequently asked questions. This process was supervised by the university's librarians, due to their expertise in the domain. The information was then extracted and a list of topics and subtopics was selected. As a result, the topics are presented as topic trees, which are topological and are presented inside an XML files. When Bob was first presented online

in 2007 the system worked using regular expression pattern matching. That is, each question submitted by a user was matched against possible answers from the topic trees, and the most probably answer was selected [14].

Bob is an interactive system, which besides allowing users to obtain information about the university library also allows for follow-up or context questions. That is, users can ask multiple questions until they are satisfied and have obtained the information that they were looking for. Users are allowed to ask follow-up questions and to have continuous dialogue. Allowing users to ask such questions is important. First, it makes the conversation more natural. Humans are used to having a conversation rather than asking questions in isolation. Second, it provides a richer environment for the user to interact in, for he/she may not be able to gain all the information they need in one question. Moreover, follow-up questions allow for the user to ask questions about things that they may not have become aware of before the interaction. For example, looking at the sample interaction in 17, we can see that a user might ask a question based on the received answer. In this example, the user asks what OPAC is, since this term was mentioned in the answer. Practically, what this means is that in order to perform well, Bob must in some way deal with these follow-up questions, as well as the way that information is carried along in the conversation between the system and the user [1].

- (17)
- (a) Can I borrow DVDs?
  - (b) If you want to borrow books, DVDs or journals you need to take them from the shelf yourself and check them out at the information desk. If you want them to be sent to Brixen-Bressanone, Bozen-Bolzano, Bruneck-Brunico or Schlanders-Silandro you can order them via the OPAC
  - (c) What is OPAC?
  - (d) What I call the OPAC is the online public access catalogue of the Library of the Free University Bozen-Bolzano. With the OPAC you can search all items kept at the Library. And you can also borrow or reserve them.

## 3.2 Logistic Regression

Researchers at the Free University of Bozen-Bolzano working on Bob had started looking at the notion of context and follow-up questions in IQA. At this point Bob had been running online using the regular expression pattern-matching architecture. This allowed for the collection of real users dialogues with Bob which could be used as training data in a machine learning framework. As such, a linear regression model was proposed by Manuel Kirschner, a PhD student at the Free University of Bolzano, working on Bob [15].



The idea behind logistic regression models (LRMs) is that to try and define the relationship between predictor features (independent variables) and an outcome. More precisely, one uses this relationship or *fit* between the predictor variables and the outcome, so that given a new set of such variables the outcome can be predicted.

In regression, the outcome variable,  $X\hat{\beta}$  can be defined as:

$$X\hat{\beta} = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n \quad (3.1)$$

in the case where there are  $n$  predictor features being used.

$\beta_0$  is the learned intercept while  $\beta_ix_i$  are the are called regression coefficients. These are the weights associated to each feature which are used to calculate the outcome.  $\beta_ix_i$  are calculated using the maximum likelihood estimate.

Using the LRM:

$$Prob(correctanswer) = \frac{1}{(1 + exp(-X\hat{\beta}))} \quad (3.2)$$

with the  $X\hat{\beta}$  defined as above, we can find the probability of a given candidate answer being the correct one.

In the case of Bob, the predictor variables can be thought of as features which in some way describe the pattern that can be seen in the discourse between the system and the user. This discourse is represented by the questions posed by the user and the answers given by the system. There is a set of possible, or candidate answers for each question which can be obtained from the topic trees (as describe above). The candidate answer with the highest probability (as demonstrated above) is then chosen as the correct one.

The Bob training data consists of bits, or snippets of question-answer interactions between the user and the system. Since the idea behind this model is to look at the role of follow-up questions and the information that they carry, the training data for Bob has been prepared in the form of four-turn snippets. Each snippet consists of the first question (Q1), the answer to the first question (A1), the follow-up question (Q2) and the answer to the follow-up question (A2). The snippets allow us to test and examine the ways that previous questions and answers (Q1 and A1) as well as the current question (Q2) can affect the retrieval of the proper answer (A2).<sup>1</sup>

---

<sup>1</sup>To see a sample of Bob snippet data see Appendix A.

In their LRM Kirschner et al. defined two types of features: A2-finding and context features. A2-finding features are the ones which take into account possible candidate answers and help facilitate the selection of the correct A2. This is done by examining the similarity between an utterance and a candidate answer. Context features are the ones that examine the role of the context (Q1, A1 and Q2 in this case) and attempt to identify it. In the LRM A2-finding features are entered as main effects while the context-defining features are used as interactions with the former.

They identified eight A2-finding features. They are based on four measures:

- (i) Lexical similarity (*lexsim*) represents the similarity of two utterances with respect to the number of shared (the same) terms.
- (ii) Distributional similarity (*distsim*) represents the similarity of two utterances with respect to both same and similar terms.
- (iii) Semantic similarity (*semsim*) represents the similarity of two utterances as measured by WordNet.
- (iv) Action sequence (*action*) which is a binary feature representing whether the two utterances are relating to the same action.

The above four measure were calculated for A1-A2 relationship (denoted as *far* features) and Q2-A2 relationship (denoted as *near*), resulting in eight A2-finding features.

Regarding the context features, in looking at Q1-Q2 pairs Kirchner et al. identified two cases: topic continuation and topic shift. Topic continuation is the case where the topic in Q2 is the same as in Q1. Topic shift is the case where the topic shifts from Q1 to Q2. The data was manually annotated for the *topshift* feature. They also attempted to automatically approximate this feature. The measures (i)-(iii) were used for this approximation. Six such features were defined: Q1-Q2.lexsim, A1-Q2.lexsim, Q1-Q2.distsim, A1-Q2.distsim, Q1-Q2.semsim and A1-Q2.semsim.

For the experiment Kirschner et al. tried several models that contained only the A2-finding features (the main effects only model) as well as using the context features together with the A2-finding features (the interaction model). The best performance was obtained using the *near*, *far* and the *topshift* features described above.

### 3.3 Context Features

As can be seen from discourse theory, work on IQA systems and Bob in particular, context plays a large role in any such systems which allows for *interaction* and for users to ask follow-up questions. That is, modeling context is something that is important and that can improve the performance of such a system. The work done on Bob by Kirschner et al. described above shows the benefit of using these context-defining features. In examining their work, there were two things I noticed that drew my attention. First, the *topic shift* feature proved very beneficial. Moreover, the manually annotated version performed better than the automatic approximation. However, manually labelling the data is both expensive and problematic for large datasets. In machine learning settings unsupervised learning is always preferred. This is especially important as the available training data grows, as is the case with Bob. The second point is that the features and the work done by Kirschner and his colleagues rely on rather shallow syntactic processing. But modeling context requires some type of *understanding* of the utterances. Therefore, I believe that deeper syntactic and semantic processing is necessary.

Taking these two considerations into account, I decided to take a deeper processing approach to modeling context for Bob. In order to do this I decided to implement the theory proposed by Chai and Jin and Sun and Chai in order to try and test whether modeling context and providing some deep analysis would be beneficial to Bob's performance. That is, using these two theories as background I extracted features that could be used by the LRM. I propose that these features will be useful for modeling context by examining the follow-up questions as well as for finding the correct A2, and can in turn be helpful in improving Bob's performance.

### 3.4 Chai and Jin

The model proposed by Chai and Jin is based on discourse roles and discourse transitions. From the theory we can see that discourse roles are based on intent, content and media. To look at media is unnecessary since Bob only provides answers in text format with a possible hyperlink. I decided not to look at discourse roles since this information is already extracted by the shallow features that are a part of the model described above. For example, the *lexsim* feature looks at the shared lexical items between two utterances.

In looking at different types of transitions between questions there are two things

which can be noticed. First, we can see the motivation of the user. That is, if there is a topic extension the user wants information that is more restrained than the information he/she obtained. That is, by adding another constraint the user is in a sense restricting the search space and the type of information that he/she wants to obtain. On the other hand, if the transition signifies a topic exploration the user wants to focus on some other part of the activity that he/she is asking about. This might signify a slight change in the direction of the information seeking, with the topic remaining the same. That is, the user is gaining information on the same topic, but wants to get a different aspect of the topic. Second, in examining the transitions we must look at the semantic entities within each sentence in order to be able to correctly classify the transition. In other words, one must extract the predicates and the arguments of each sentence in order to do this classification.

In a sense discourse transitions cover both content and intent, since doing the above classification requires the examination of topic and focus for each sentence. In order to examine and find such transitions, one must look at their content in order to see if there has been a change in topic or focus. Moreover, the different types of transitions inherently portray the intent of the user, and how this intent might be changing throughout the discourse.

The work done on Bob by Kirschner et al. as well as the evidence from Dialogue theory supports the need to model context. The linear regression model for Bob mentioned above allows for the modeling of context by looking at relations between two consecutive questions (Q1 and Q2). However, all these features are not semantically rich and they require rather shallow processing. The theory of Chai and Jin requires deeper processing that examines the syntax and semantics of the dialogue. Therefore, I decided to extract five context-defining features from this theory, based on the five possible transitions: constraint refinement, participant shift, topic exploration, topic activity shifts to topic activity and topic activity shifts to topic entity.

### **Syntactic Processing**

As mentioned above, the extraction of the features based on the theory by Chai and Jin requires deep syntactic and semantic processing. Therefore, the first step for me was to perform the necessary processing in order to be able to do the transition classification.

In order to be able to examine the questions and detect the different transitions, I decided to extract the predicate-argument structure for each sentence, as well

as the question type, or *wh*-word for each question. The *wh*-question signifies what type of information the user is seeking. The set of possible *wh*-words include *what*, *when*, *where*, *why*, *how*, *who*, and *which*. I extracted the *wh*-word for each question based on simple string patten-matching. In the case where there was no *wh*-word present, it was set to be the empty string.

- (18)
- (a) Who can use the library?
  - (b) And where can I find books?
  - (c) Can I order documents from abroad?

In example 18, the extracted *wh*-words for 18(a), 18(b) and 18(c) are *who*, *where* and the empty string, respectively.

### Stanford Parser

In order to examine the different types of transitions, one must be able to extract what the topic and focus of a given utterance are. In order to do this, one must look at the predicate and arguments of the given utterance. For the extraction of the predicate-argument structure I used the Stanford Parser. [17] The Stanford Parser is one of the most widely used parsers for English, which on average achieves quite a high parsing accuracy. It produces a phrase structure parse for a given sentence. However, the Stanford Parser can also generate typed dependency parses. This type of parse shows relations between pairs of words. The Stanford Parser produces typed dependencies, meaning that such relations are labelled. That is, there are 49 defined dependencies which are defined. Dependencies include relations such as *nominal subject* (or *nsubj*) or *adjectival modifier* (or *amod*). [7, 8, 6]

- (19)
- (a) *nsubj*(talks, Mary)
  - (b) *amod*(book, big)

The dependency relations are all bigrams, meaning that they show the relationship between two words. For example, looking at example in 19(a), we know that *Mary* is the subject of the verb *to talk*. In 19(b), we can observe that we have the phrase

emphbig book, that is *big* is the adjectival modifier of *book*.<sup>2</sup>

---

<sup>2</sup>To see full list of Stanford Parser Dependencies see Appendix B.

I therefore used the typed dependency parsing, as provided by the Stanford Parser to extract the predicate and arguments for each question. In this way I could use the dependencies and their arguments to extract the predicate and the different arguments, such as subject and object.

In example 20 below, there are three examples of questions from the Bob data and their corresponding parses. For each sentence there is the phrase structure parse as well as the dependency parse.

(20)

(a) Where can I find computers?

(ROOT (SBAR (WHADVP (WRB where)) (S (NP (FW i)) (VP (MD can) (VP (VB find) (NP (NNS computers))))))))

[advmod(find-4, where-1)  
nsubj(find-4, i-2)  
aux(find-4, can-3)  
dobj(find-4, computers-5)]

(b) I need the full text of an article.

(ROOT (S (NP (FW i)) (VP (VBP need) (NP (NP (DT the) (JJ full) (NN text)) (PP (IN of) (NP (DT an) (NN article)))))))

[nsubj(need-2, i-1)  
det(text-5, the-3)  
amod(text-5, full-4)  
dobj(need-2, text-5)  
det(article-8, an-7)  
prep\_of(text-5, article-8)]

(c) yes

(ROOT (INTJ (UH yes)))

□

For each sentence, I extracted the following information (if present): the predicate, the subject, the subject modifier, object(s), the object modifier(s), prepositions and their arguments, temporal arguments and adverbial modifiers. The predicate, subject and object are the three arguments which are explicitly named. The rest are called arg0, arg1, arg2, etc.

For example 20(a), the follow information is extracted:

*Predicate:* find

*Subject:* I

*Object:* computers

The *wh*-word is not extracted using the parse, since we separately extract and consider this word. For each of the prepositions, the preposition and the words which are its arguments are remembered. For example, in 20(b), for the phrase text of article, the argument is article, the preposition associated with it is of and it is attached to the object text.

For example 20(b), the following information is extracted:

*Predicate:* need

*Subject:* I

*Object:* text

*Arg1:* article, with preposition of, attached to the object text

There are some instances where no predicate-argument structure can be extracted, such as in example 20(c).

### **Normalization of Data**

The questions that are presented to Bob are specified by the user. Users will often use short forms such as *isn't* for *is not*, or *there's* for *there is*. Therefore, it is useful to do some type of data clean-up or normalization.

OpenEphyra is an open-source question-answering system. [16] OpenEphyra is an example of a classical QA or IQA system: it has all the components described in section 1.3. The system is open source and can be quite beneficial for building an IQA system. While OpenEphyra is mainly designed to handle non-interactive QA, the thing that I found really valuable is that it contains many tools available for the different stages of IQA such as question analysis and answer extraction. [22] The system also has the option of using the Stanford Parser for the parsing of the questions, which I found useful since I use this parser to extract the predicate-argument structure.

I decided to use the question analysis part of the OpenEphyra system to do the normalization of the data. First,I used the question normalizer. The normalizer does three things. First, it gets rid of leading, trailing, and multiple whitespaces. Second, it drops the filler words. Filler words are words such as *approximate* or *so-called*, which do not add any semantic information to the

sentence. Finally, the normalizer converts short forms such as *there's* and *isn't* to their non-concatenated form.

(21)

(a) What's refworks?

```
[attr('s-2, what-1)
 nsubj('s-2, refworks-3)]
```

(b) What is refworks?

```
[attr(is-2, what-1)
 nsubj(is-2, refworks-3)]
```

In example 21, 21(a) shows a non-normalized question and 21(b) shows the normalized version. We can also see the different parses that are produced for the two different questions- the non-normalized and the normalized versions. The parse for 21(b) is more correct since the predicate of the question is *is*, not *'s*. This is especially important in cases where the two predicates of two questions are being compared and one has the form *'s* and the other *is*.

Second, I used the verb modifier module from OpenEphyra. The verb modifier performs two tasks. First, it gets rid of punctuation marks, including not only periods and question marks but quotations, brackets, etc. Second, it looks at cases where the verb is made up of an auxiliary and a gerund or past participle. What the modifier does is change the word order in cases where there is some word that occurs between the auxiliary and gerund or past participle. It changes it in such a way as to make sure that the auxiliary is immediately followed by the gerund or past participle. This type of change happens very often in questions, due to the insertion of *wh*-words which are usually then moved to the front of the sentence.

(22)

(a) And where can I find books?

```
[advmod(find-5, where-2)
 aux(find-5, can-3)
 advmod(find-5, i-4)
 rcmnd(and-1, find-5)
 dobj(find-5, books-6)]
```

(b) and where i can find books



```

[advmod(find-5, where-2)
 nsubj(find-5, i-3)
 aux(find-5, can-4)
 dep(and-1, find-5)
 dobj(find-5, books-6)]

```

Example 22 shows the difference between a sentence where the verb modifier has been used 22(b), compared to the non-modified form 22(a). From this example, one can see how this change in order can affect the parse. The parse of 22(b) clearly identifies *I* as the subject of the predicate *find*. The parse in 22(a) finds no subject for this phrase, which is clearly wrong.

After having normalized the given questions for each of the snippets, I extracted the predicate and arguments for each of them. I used this extracted information to examine pairs of consecutive questions for the presence of the five different transitions as defined by Chai and Jin.

### 3.4.1 Constraint Refinement

Constraint refinement, as defined above, is the transition where both the questions have the same topic, but the second question has additional constraints.

Since this transition examines the case where the two questions have the same topic, I defined this transition as the case when the two consecutive questions have:

- (i) The same *wh*-word (i.e. they have the same focus)
- (ii) The same predicate (i.e. they are talking about the same activity)
- (iii) Q2 has an argument that was not present in Q1 (i.e. the new constraint)

(23)

- (a) Can I borrow all of them?

```

[nsubj(borrow-3, i-1)
 aux(borrow-3, can-2)
 dobj(borrow-3, all-4)
 prep_of(all-4, them-6)]

```

- (b) Can I borrow them online?

```

[nsubj(borrow-3, i-1)
 aux(borrow-3, can-2)

```

```
dobj(borrow-3, them-4)
amod(them-4, online-5)]
```

Example 23 is an example of constraint refinement found in the Bob data. We can see that both the questions have the same predicate (*borrowing*) and the same type of question (neither have a *wh*-phrase). However, the constraints are different for the two questions. In question 23(b), we have the constraint *online* that is not found in 23(a).

### 3.4.2 Participant Shift

Participant shift is defined as the case when the two questions are talking about the same topic, but have different participants. I defined this transition as the case when two consecutive questions have:

- (i) The same *wh*-word (i.e. they have the same focus)
- (ii) The same predicate (i.e. they are talking about the same activity)
- (iii) An argument found in both Q1 and Q2 has a different value for the two sentences (i.e. the different constraint)

Since the transition looks at shift in participants, I only examine a change in subject, object or the argument of given preposition.

(24)

(a) Where can I find books about library science?

```
[advmod(find-4, where-1)
nsubj(find-4, i-2)
aux(find-4, can-3)
dobj(find-4, books-5)
nn(science-8, library-7)
prep_about(find-4, science-8)]
```

(b) Where can I find information about the library?

```
[advmod(find-4, where-1)
nsubj(find-4, i-2)
aux(find-4, can-3)
dobj(find-4, information-5)
det(library-8, the-7)
prep_about(find-4, library-8)]
```

In example 24, we see an example of participant shift in the Bob data. Both the questions have the same *wh*-word (*where*), and the same predicate (*find*). The difference is that in 24(a) the user is asking about finding *books*, while in 24(b) the user is asking about finding *information*.

### 3.4.3 Topic Exploration

Topic exploration classifies the transition where the two questions share the same topic, but are talking about different aspects of it. In other words, this is the case where the two questions are talking about the same activity (i.e. have the same predicate), but have a different focus. I classify this transition as the case where the two consecutive questions have:

- (i) The same predicate and subject (i.e. the same activity)
- (ii) Different *wh*-words (i.e. different focus)

In this case, as with the other transitions one must be careful in instances where there is a pronoun in the second question. Therefore, in comparing arguments such as subject and objects I take into account the possibility that one of them might be a pronoun, which would signify that it is the same entity as the one in the previous question.

(25)

(a) And can I print?

```
[dep(and-1, can-2)
 nsubj(print-4, i-3)
 rcmod(can-2, print-4)]
```

(b) Where can I print?

```
[advmod(print-4, where-1)
 aux(print-4, can-2)
 nsubj(print-4, i-3)]
```

Example 25 is one of the cases of topic exploration found in the Bob data. Both the questions are talking about *printing*. However, the first question, 25(a), is asking *if* one can print, while the second 25(b) is talking about *where* one can print.

### Activity versus Entity

The next two transitions are based on the notion of entity and activity. Intuitively one can think of activities as questions that talk about actions one can perform such as *borrowing*, *ordering* or *searching*—. Entities, on the other hand are questions that ask about specific objects such as *books*, the *library* or the *library catalogue*.

(26)

- (a) When can I return a book?
- (b) What is OPAC?

26(a) is an example of an *activity* question, while 26(b) is an example of an *entity* question.

I have defined activity questions as ones that have a verb other than the verbs *to be*, *to have* or *to do* (in any form). The activity that this question is centered on is the predicate itself. Conversely, questions that only contained the above mentioned verbs (or no verb at all), are entity questions.

### MiniPropBank

For the topic shift features, on top of classifying questions as activity or entity based one, I needed to be able to see whether the two activities are related, or whether the entity of Q2 is related to the activity of Q1. For this I have used a MiniPropBank that has been specially designed for the Bob data. The MiniPropBank contains a list of the most common verbs or activities that are found in the Bob data that was collected online. Each of these verbs has three things listed for it. First, there is the list of other verbs with the same meaning (i.e. the synonyms). For example, instead of the verb *to search* one can also use *to browse*, *to seek*, *to find*, *to research* or *to look for*. Second, each verb has a list of related activities. For *search* this includes the verbs *borrow*, *reserve*, *order*, *request acquisition* and *emphphotocopy*. Finally, there is a list of named entities associated with this activity. For *search* this includes *information desk*, *website*, *catalogue*, *field*, *for user*, *website* and *user*.<sup>3</sup>

#### 3.4.4 Topic Activity shifts to Topic Activity

This transition is represented by two consecutive questions about two different but related activities. I defined this feature as the case when:

---

<sup>3</sup>To see an example of a task entry in the MiniPropBank file see Appendix C.

- (i) Both are activity questions
- (ii) The predicates that define each of the two questions (or their synonyms) are related activities, as defined in the MiniPropBankFile.

Therefore, in order to extract this feature I look for pairs of activity question. If this is the case, I extract the predicate for each question. Finally I check whether one of the predicates is in the MiniPropBankFile with the second predicate listed as its related task.

(27)

- (a) How can I borrow a book?

```
[advmod(i-2, how-1)
dep(borrow-4, i-2)
aux(borrow-4, can-3)
det(book-6, a-5)
dobj(borrow-4, book-6)]
```

- (b) How can I order a book?

```
[advmod(i-2, how-1)
dep(order-4, i-2)
aux(order-4, can-3)
det(book-6, a-5)
dobj(order-4, book-6)]
```

The above example, 27, shows the activity topic to another activity topic transition. Both of the questions are activity questions. 27(a) concerns itself with the activity of *borrowing* and 27(b) with the activity of *ordering*. Since these two activities are defined as related verbs in the MiniPropBank file, this is an example of such a transition.

### 3.4.5 Topic Activity shifts to Topic Entity

This transition is defined as the case when the first question talks about an activity and the second question talks about an entity related to the activity in the first question. I defined this transition as the case when:

- (i) The first question is an activity question and the second question is an entity question
- (ii) The second question contains one of the tasks associated with the predicate defining the activity question

That is, in the case of the first question being an activity question and the second an entity one, I extracted the predicate from the first one. I looked up this predicate in the list of tasks in the MiniPropBank file to obtain a list of associated entities. If any of these entities were present in the second question than this feature is true.

(28)

- (a) Where should I return the book?

```
[advmod(return-4, where-1)
aux(return-4, should-2)
nsubj(return-4, I-3)
det(book-6, the-5)
dobj(return-4, book-6)]
```

- (b) Where is the information desk?

```
[advmod(is-2, where-1)
det(desk-5, the-3)
nn(desk-5, information-4)
nsubj(is-2, desk-5)]
```

Example 28 shows such a transition in the Bob data. 28(a) is an activity question relating to the predicate *return*. One of the entities related to this task as defined in the MiniPropBank file is *information desk*, which is found in 28(b).

### 3.5 Sun and Chai

One of the issues that comes up in the work of Chai and Jin, as well as Dialogue theory is that of anaphora. In dialogue or in IQA interactions it is normal to expect that one will encounter anaphora. In order to be able to grasp the full meaning of an utterance containing an anaphora, one must be able to gain access to its antecedent. The model proposed by Sun and Chai is based on being able to do anaphora resolution. Further, it is based on being able to find the centers, or the main semantic actors of a given utterance (or questions in

this case). Therefore, in order to implement this model one must be able to resolve pronouns and find their antecedents, and then find the centers of the given utterances. In order to do this, I used a tool named GuiTAR.

### GuiTAR

GuiTAR is an anaphora resolution tool that is useful for not only doing pronoun and anaphora resolution, but for finding centers of utterances as well. [11] I used GuiTAR version 3.0.3. GuiTAR is built on the Charniak parser. That is, GuiTAR requires that data be parsed and tagged by the parser in order to be able to do the resolution. The Charniak parser is a probabilistic context-free grammar parser, which is very popular for English due to its high parsing accuracy.

The Charniak parser, as well as GuiTAR require for the input file to be a text file with a list of sentences, one on each line. In order to maintain the order of sentences in the Bob data file I made sure that each utterance consisted of one and only one sentence. This is because the parser and GuiTAR create one node for each sentence in the file. This means that I removed periods, exclamation marks and question marks and replaced them by commas. This does not affect the outcome of the algorithm since this change does not affect the semantics of the sentences (i.e. the named entities inside each sentence will stay the same). Since each snippet of dialogue contains two questions(Q1 and Q2), the input file contained 3044 sentences.

The output of the Charniak parser is an XML file, with each sentence being presented by a node. Further, each sentence contains all the words, which are tagged with the appropriate part of speech. Nouns in particular are marked for their category, person, number, and gender.

(29)

(a) Noun: *I*

Category: personal pronoun

Person: first

Number: singular

Gender: any

(b) Noun: *books*

Category: np

Person: third

Number: plural

Gender: neutral

In (29) we can see two examples of nouns and their marked category. This information is important since when we try to resolve a pronoun to its antecedent, these categories are used in order to find the correct match. For example, if we have the pronoun *it*, its antecedent cannot possibly be *books*, since these two words do not match in terms of their number.

I used the shell script *preprocCHGuitar.sh* that is provided with GuiTAR along with the Charniak parser in order to obtain the tagged output file. I obtained this file by running the command:

```
preprocCHGuitar.shsentences
```

where *sentences* is the text file containing the sentences to be processed. Running this command outputted a file called *tagged.sentences.xml*.

Once all the given utterances have been parsed and marked with the specified information the resolution algorithm can proceed. GuiTAR proceeds with the following steps. First, each named entity is given a unique id. This unique id is important so that we can find a way to access an antecedent for a given anaphora. Next, the algorithm looks for anaphora in the utterances and finds their antecedents based on the extracted syntactic information. The information is presented inside an XML file. In the XML file, there is a node for each sentence. Inside each sentence there is a node for each anaphora that has been found and resolved (if any have been found). This node contains the id of the anaphora and the id of the pronoun and of the antecedent.<sup>4</sup>

Using the downloaded GuiTAR package as well as the *convert.jar* file that is provided along with the package, I obtained the processed file by running the following command:

```
java -cp gtar3.0.3.jar : trove.jar : googleapi.jar : libsvm.jar : . GTAR_main  
-preprocess -log -t pennTagSet.ini -verbose -i tagged.sentences.xml
```

where *tagged.sentences.xml* is the file outputted by the Charniak parser. Running this command produced the desired *processed.tagged.sentences.xml* file which contains the anaphora resolution and the classified named entities.

---

<sup>4</sup>To see a sample of a sentence processed by GuiTAR see Appendix D.



## Finding the Centers

I used GuiTAR to parse and resolve all the pairs of consecutive questions found in the Bob data. There are two types of named entities that must be considered for each sentence. First there are the non-anaphoric named entities and second there are the anaphoric named entities. For the former ones we can access their meaning from the words themselves. For the latter case, in order to obtain the meaning we must access the correct antecedent. I used the XML file produced by GuiTAR to do two things. First, to see whether for each given question has an anaphora. If there is an anaphora I kept note of this and the antecedent for all the anaphora, for each given question. Second, each question that is processed by GuiTAR has its named entities tagged. Therefore, I was able to get a list of all the named entities for each given question.

The centers of a given utterance are none other than the semantic entities. Therefore, the centers of any given question are the named entities that have been found and tagged by GuiTAR. For each given question I extracted and kept track of all the named entities that are found in it. Moreover, I kept note of the entities that come from anaphora (that is their antecedents) and those that do not.

In examining the Bob data, I noticed that there are many instances of personal pronouns being used, especially *I*. This is understandable since the user in interacting with the system is very often asking questions about what he/she wants to know or needs. However, these personal pronouns are rarely found in the answer candidates. As such, I decided to ignore the personal pronouns as possible centers for sentences that contain other named entities.

### 3.5.1 Reference Model

The reference model is the most basic one. It is based on adding the keywords of resolved pronouns to the current query. However, I did not perform a query search, but rather needed to extract features based on the described models. Therefore, I defined a follow-up question as being positive (i.e. having feature value one) for the reference model feature when:

- (i) Q2 has an anaphora contained in it (one or more)
- (ii) The antecedent to the anaphora in Q2 can be found in the candidate answer

Intuitively, a question has the reference model feature being positive if the use of the antecedent is useful in finding the correct answer, meaning that this word

is contained in the answer. This is an example of an A2-finding feature, since different values are assigned to different answer candidates.

(30)

(Q1) How many floors has the library?

(A1) [...]

(Q2) Are there computers?

(A2) [...]

Looking at example 30, as taken from Bob data, we have an example where the *there* in Q2 has to be resolved and its antecedent found. Its antecedent in this case is the word *library* (found in Q1). Therefore, the reference model is positive (i.e. has a value one) for this Q2 and all the candidate answers that contain the word *library*. Conversely, the feature for this model is negative (i.e. has value zero) for all the candidate answers that do not contain the word *library* in them.

### 3.5.2 Forward Model

The forward model goes one step further than the reference model. In the forward looking model, we add the forward looking centers of Q1 to the query expansion of Q2. Therefore, I defined the feature forward model by comparing the forward looking centers of Q1 to the possible candidate answers of Q2 (or A2). That is I define this feature as the case where:

(i) Set S denoted the set of centers of Q1 (both anaphoric and non-anaphoric named entities)

(ii) A named entity from set S is found to be present in candidate answer

(31)

(Q1) Where can I find computers?

(A1) [...]

(Q2) Can I print?

(A2) [...]

Looking at Q1 in 31, the forward looking centers are *I*, *computers*. Therefore, the forward model is positive for all the snippets in which A2 contains the word *computer*. The center *I* is not considered, for the reasons mentioned above.

### 3.5.3 Transition Model

The transition model builds both on the reference and the forward models. For this model I needed the preferred entity for both Q1 and Q2. This means that

for questions that have more than one center I needed to find the preferred centre. Priority is first given to anaphora. That is, for questions that have one or more anaphora the entity that the anaphora is referring to is given priority, and is therefore selected as the preferred centre. In the case of multiple anaphora linear order is used with the first one found being the preferred one. In the case that there is no anaphora, linear order is used as well, with the entity that comes first being selected as the preferred entity.

I extracted the preferred centre for Q1 and Q2. I then classified the transitions accordingly. If the two entities have the same head and modifier, than this is transition type one, or *continue*. If the head is the same but the modifier is different, than this is transition type two, or *retain*. If the modifier is the same but the head is different than this is transition type three, or *smooth shift*. Finally, if both the head and modifier are different, than this is transition type four, or *rough shift*.

Very often the preferred centers only contain a head and no modifier. I note that for the *continue* transition I treated two centers that have the same head and no modifier as being the same entity. Further, for the rough shift transition I treated two centers with different heads and no modifier as being two completely different entities.

- (32)  
(Q1) How many books are in the library?  
(A1) [...]  
(Q2) Who can borrow books from the library?  
(A2) [...]

This is an example of the *continue* transition. That is both Q1 and Q2 are talking about the same entity: *book*.

- (33)  
(Q1) Does a search return hits in all languages?  
(A1) [...]  
(Q2) What does the multilingual search option do?  
(A2) [...]

This is an example of the *retain* transition. That is, the named entity in Q1 is *search*, while in Q2 it is *multilingual search*. The heads are the same, while the modifiers are different (more precisely Q2 has a modifier for the head and Q1 does not).

(34)

(Q1) What is OPAC?

(A1) [...]

(Q2) How many books can I borrow?

(A2) [...]

This is an example of the *rough shift* transition. The two heads are different (*OPAC* for Q1 and *book* for Q2) and there is no modifier.

(35)

(Q1) Library password?

(A1) [...]

(Q2) Library number?

(A2) [...]

This is an example of the *smooth shift* transition. The two heads are different (*password* for Q1 and *number* for Q2), while the modifier is the same (*library* in both cases).

## Chapter 4

# Evaluation and Discussion

### 4.1 Evaluation

#### 4.1.1 Data

The training data that I used for the experiments was collected from actual on-line user's interactions with the system. The data was manually cleaned. For example, single questions were removed, as well interactions that were deemed problematic. From the data four-turn snippets were extracted. That is, that data was divided into snippets that contain a question (Q1), followed by the answer to this questions (A1), and followed by the next questions (Q2 or follow-up questions-FUQ). Only snippets where the answer to the first questions (A1) was deemed correct were used. This was done in order to avoid FUQs that were affected by wrong answers.

One thing that needs to be addressed in not only Bob, but any IQA system is the notion of a correct answer. In order to test the performance of any IQA system one must first define what constitutes a correct answer. In the case of factoid questions such as *What are the opening hours of the library?*, the answer is fairly simple. However, some questions require a much more complicated an elaborate answer. For example, the question *How do I search for economics books?* can have multiple answers depending on whether the user wants to search for the books online or at the library.

In the data I used for all the three-turn interactions the correct A2 was obtained by asking the librarians to pick the correct one. As previously mentioned, only four turn snippets were used since there is evidence that most of the information relating to a FUQ can be found in the immediately preceding answer (A1) and

question (Q1).

In the end the data consisted of 1522 four-turn snippets. It should also be noted that the set of possible answers at the time of the experiments contained 306 possible candidates.

### **Feature Extraction**

I used the 1522 snippets and 306 candidate answers to extract the features supported by Chai and Jin’s and Sun and Chai’s theories as described in Chapter 3. In total, I extracted nine features. There are two types of features that I extracted: ones that relate Q1 and Q2 and ones that try and help predict the correct A2.

### **A2-Finding Features**

The A2-finding features are the features that try and predict the correct A2. I extracted two such features: the reference model feature (*Center.Ref*) as well as the forward model feature (*Center.Forward*), as described by Sun and Chai’s work. These two features look at all the answer candidates, and assign a value of one if this feature is present between the given snippet and the answer candidate.

### **Context-defining Features**

The remaining set of seven features do not take into account the correct or possible A2s at all. Rather, these features look at the possible relationship(s) between a question and its follow-up question (Q1 and Q2). Therefore, these features are not A2-finding features, but rather features that attempt to define the context of the interaction. The seven context defining features that I extracted are: Transitions model (*Center.Transitions*), constraint refinement (*Chai.Co-Ref*), participant shift (*Chai.P-Sh*), topic exploration (*Chai.T-Ex*), activity to activity shift (*Chai.A-A*) and activity to entity shift (*Chai.A-E*).

I stored the results of the feature extraction I performed in a table. Each row contains the values of the nine extracted features, as well as the snippet ID and answer candidate ID. The table contains 465732 rows, for all the possible combinations between the 306 answer candidates and 1522 snippets.

## **4.2 Results**

I used the LRM described in section 3.2 for the evaluation [15]. I tested to see how the addition of my features would work with the LRM. The A2-finding

features were tested alone as well as within a more complex model. The context defining features were tested using the more complex, interactive model.

### Main Effects Only Models

For the A2-finding features the baseline is the case where A2s are picked at random. The best single feature result was obtained from the *near.distsim* feature.

Model Name	Mean Rank
<i>Baseline</i>	148.4678
<i>Center.Reference</i>	153.8072
<i>Center.Forward</i>	149.3702
<i>near.distsim</i>	59.5611

Table 4.1: Mean rank of correct A2 out of 306 answer candidates in single feature main effects only models.

Table 4.1 shows the results of running the two A2-finding models as single features, compared to the baseline and best single models. We can see that both of these models perform worse than the baseline. This means that these two models alone are worse at picking the correct A2 than the baseline, meaning that they fail to find the correct A2s.

### Models with an Interaction

The next test was performed to see how adding the A2-finding features together with the same context (or interaction) features which produced the best performing model by Kirschner et al. The best performing model from Kirschner et al.’s work is:

$$\begin{aligned}
 \text{answerCorrect} \sim & A1.Q2.distsim + far.lexsim + near.lexsim + near.distsim + \\
 & far.distsim + far.semsim + near.action + A1.Q2.distsim * near.distsim + \\
 & A1.Q2.distsim * near.semsim + A1.Q2.distsim * far.action \quad (4.1)
 \end{aligned}$$

The best model amongst those which do not contain any context or interaction features as found by Kirschner et al is:

$$\begin{aligned}
 \text{answerCorrect} \sim & far.lexsim + near.lexsim + far.distsim + near.distsim + \\
 & far.semsim + near.semsim + far.action + near.action \quad (4.2)
 \end{aligned}$$

In performing a fast backwards elimination procedure as part of the LRM, the *Center.Forward* feature was dropped since it did not prove to help the model’s

performance in any way.

The following formula represents the resulting model:

$$\begin{aligned} & \text{answerCorrect} \sim A1.Q2.distsim + \text{near.action} + \text{near.distsim} + \text{Center.Reference} + \\ & \text{near.lexsim} + \text{far.semsim} + \text{far.lexsim} + \text{far.distsim} + \\ & A1.Q2.distsim * \text{near.distsim} + A1.Q2.distsim * \text{far.action} \quad (4.3) \end{aligned}$$

Next, one by one each of the context-defining or interaction features was added to the model described in 4.3. Then the backward elimination procedure was performed, which drops features which are not informative (i.e. they do not contribute to the model).

Adding the *Center.Transition* feature the resulting model:

$$\begin{aligned} & \text{answerCorrect} \sim \text{Center.Transitions} + \text{near.action} + \text{near.distsim} + \text{Center.Ref} + \\ & Q1.A1.lexsim + \text{tfIdf}..Q2.A2 + \text{far.action} + \text{far.lexsim} + \\ & \text{Center.Transitions} * \text{near.distsim} + \text{Center.Transitions} * Q1.A2.lexsim + \\ & \text{Center.Transitions} * \text{far.semsim} + \text{Center.Transitions} * \text{far.action} + \\ & \text{Center.Transitions} * \text{far.lexsim} \quad (4.4) \end{aligned}$$

Adding the *Chai.P-Sh* feature, there is no model produced since the feature is eliminated during the backward elimination procedure.

Adding the *Chai.Co-Ref* feature the resulting model is:

$$\begin{aligned} & \text{answerCorrect} \sim \text{Chai.Co-Ref} + \text{near.action} + \text{near.distsim} + \text{Center.Ref} + \\ & \text{near.lexsim} + \text{far.semsim} + \text{far.action} + \text{far.lexsim} + \text{Chai.Co-Ref} * \text{near.distsim} + \\ & \text{Chai.Co-Ref} * Q1.A2.action + \text{Chai.Co-Ref} * \text{far.lexsim} \quad (4.5) \end{aligned}$$

Adding the *Chai.T-Ex* feature, there is no model produced since the feature is eliminated during the backward elimination procedure.

Adding the *Chai.A-A* feature, there is no model produced due to numerical problem in calculating the LRM.

Adding the *Chai.A-E* feature, there is no model produced due to numerical problem in calculating the LRM.

Table 4.2 show the means of the three models that were found using my fea-



Model	Mean	SD
<i>refnoi</i>	50.35381	69.0084
<i>refkbest</i>	44.22503	64.1189
<i>refbest</i>	43.97273	63.91284
<i>refct</i>	48.36629	67.69906
<i>refcor</i>	49.33804	67.99789

Table 4.2: Mean ranks of correct A2 of out 306 answer candidates in interaction models.

tures as well as the two models mentioned from Kirschner et al.’s work. Model 4.3 is the best model since it has the lowest mean. However, the difference between this model and model 4.1 is not significant (using paired t-test with significance level  $p \leq 0.05$ ). What this shows is that this model is a good A2-finding model and is on the same level as the best model found by Kirschner et al.

The 4.5 and 4.4 models perform worse than the best model. However, if we compare the results to those of the model without interactions (i.e. model 4.2), we find that the models do perform better. The difference between model 4.5 and model 4.2 is significant (Paired t-test,  $t = 2.6455$ ,  $df = 1521$ ,  $p - value = 0.00824$ ). Also, the difference between model 4.4 and model 4.2 is significant (Paired t-test,  $t = 3.9847$ ,  $df = 1521$ ,  $p - value = 0.000075$ ). The difference between model 4.4 and model 4.5 is not significant.

What this shows is evidence that the *Center.Transitions* and *Chai.Co-Ref* features provide context modeling which proves beneficial. The other features which are dropped from the model since they are not beneficial and do not provide any context modeling using the given data.

### Illustrative Example

The model shown in Table 4.3 is a rather complex one that requires the values of many different features as well as their corresponding coefficients. To illustrate how a LRM and the features I have extracted are used to find the correct answer I will provide an example with a simpler model.

The simpler model, consisting of the three Centering features I extracted:

$$\begin{aligned}
& \text{answerCorrect} \sim \text{Center.Forward} + \text{Center.Ref} + \\
& \text{Center.Transitions} * \text{Center.Forward} + \text{Center.Transitions} * \text{Center.Ref}
\end{aligned}
\tag{4.6}$$

Predictor	$\beta$ coefficient	St.Error	z-value	$Pr(>  z )$
Intercept	-5.764712	0.11956	-48.22	0.0000
Center.Forward	1.186541	0.28488	4.17	0.0000
Center.Ref	-1.257909	0.27797	-4.53	0.0000
Center.Transitions	-0.007585	0.03240	-0.23	0.8149
Center.Forward * Center.Transitions	-0.254154	0.07464	-3.41	0.0007
Center.Ref * Center.Transitions	0.329925	0.07597	4.34	0.0000

Table 4.3: The predictors for model 4.6.

The model in 4.6 is based on the three Centering features (namely *Center.Transitions*, *Center.Ref* and *Center.Forward*). Table 4.3 shows the  $\beta$  values that are calculated for this model.

- (36)
- (Q1) How many books are in the library?.
- (A1) [...]
- (Q2) Can I borrow all of them?
- (A2) [...]

Example 36 is a snippet taken from the Bob data. Let us consider two possible candidate answers:

- (CA1) Some items are reference only, for example material for the reserve collection or reference books. Those items have a red sticker and in the OPAC you can see the information reference library.
- (CA2) How can I download the article from metalib?

For both the candidate answers the *Center.Transitions* feature is equal to one, since the preferred center of both Q1 and Q2 is *books*. For CA1 for both the *Center.Ref* and *Center.Forward* features the value is one. This is because CA1 contains the word *books* and *library* which are the centers of Q1 and Q2. For CA2 for both the *Center.Ref* and *Center.Forward* features the value is zero. This is because neither *books* nor *library* is contained in CA2.

Using the formula and model above, we get:

$$P(\text{CA1 is correct} | \text{CA2}) = 1 / (1 + e^{(1.186541)1} + (-1.257909)1 + (-0.254154)1 * 1)$$

$$1 + (0.329925)1 * 1) = 0.501$$

$$P(CA2iscorrectA2) = 1/(1 + e^{(1.186541)0 + (-1.257909)0 + (-0.254154)1 * 0 + (0.329925)1 * 0}) = 0.5$$

Therefore, using these two candidate answers and the given model and formula, we would select CA1 as the more probably candidate answer. The model described above is much simpler than the models described above whose results were found to be significant. The model is used to illustrate how the work described above is used to help predict the correct answer.

### 4.3 Discussion

There are several things to be noticed from the evaluation performed above. First, while no A2-finding feature was good for finding the correct answer on its own, one of these models in combination with other features proved to obtain one of the best results. The benefit of the *Center.Ref* model portrays the importance that anaphora resolution plays in IQA. In order to fully and automatically understand an anaphoric utterance it is beneficial, if not necessary to be able to access and interpret its antecedent. The fact that the *Center.Forward* model was not beneficial points towards the idea that the centers of the previous utterance are much more important than those of the utterances which can be found further back in the dialogue.

It is interesting that out of the five context features presented by Chai and Jin’s theory, only one of them was found to be beneficial in modeling context, namely the *Chai.Co-Ref* feature. This feature is a type of topic continuation feature. What these findings suggest is that context is especially important in instances where there is a continuation of topic from question to question. This makes sense, since topic continuation signifies that there is a lot of information in common between the two questions. There are other possible explanations, such as the sparseness or particularity of this data (both of which are described below).

The results relating to the *Center.Transitions* and *Chai.C-Ref* features show how context-defining features can be beneficial. This is evident from the fact that the models which include these features perform significantly better than the same model without the context feature (i.e. having only the A2-finding features). This supports the idea that context in IQA is important and that it carries valuable information. It further supports the mentioned theories which attempt to model context in the hope that they will be beneficial to IQA sys-

tems.

The evaluation also shows us the need for using both A2-finding as well as context-defining features. The results are best when a mix of these two features is used. Further, the best results is produced using a mix of both *shallow* and *deep* features. What these two findings suggest is that the best results are produced using a mix of techniques and representations.

The fact that the feature extraction process is done automatically is important. In the model proposed by Kirschner et al., one of the features, that is the presence or absence of a topic shift is manually coded. This is very time consuming and also very inconvenient for machine learning approaches which usually try and work with large training sets if possible. With Bob this training set is constantly growing since the system is available online and new data is constantly being collected. Features and processing that is done automatically is therefore preferred. The work presented above does exactly this and shows how semantic and syntactic information can be automatically extracted.

Finally, the evaluation above shows the need to try and test more models based on a mix of different features. The work above shows that the best results are achieved using a mix of features and a mix of types of features. Work should be continued, with more models and more features being proposed.

### 4.3.1 Issues to Consider

There are several issues that should be addressed in discussing the evaluation results. Below is a survey of these issues.

#### Sparseness of Representative Data

Looking into the number of positive instances extracted for all the features (Table 4.4, Table 4.5 and Table 4.6) we can see that there is the issue of sparseness of data. That is, it is obvious that the reason a LRM could not be calculated for *Chai.A-A* and *Chai.A-E* is that there are so few instances. There are only three and four such positive instances, respectively. The question is whether this type of transition is really rare or whether there was a problem with the method used to extract the feature. One of the possible problems could be that the MiniProbBank file might not contain all the related activities and entities. A future development could include using WordNet instead of the MiniPropBank.

For the rest of the features there does not seem to be a problem with sparseness

of representative data. The *Chai.Co-Ref*, *Chai.P-Sh*, and *Chai.T-Ex* features all seem to contain quite a few positive instances. Also, while transitions two and three are quite rare for the *Center.Transitions* feature, the feature always has a value and therefore this does not appear to be a problem. This is supported by the fact that this feature was found to facilitate context modeling as well.

Feature	Num of found positive features
<i>Chai.Co-Ref</i>	412
<i>Chai.P-Sh</i>	150
<i>Chai.T-Ex</i>	285
<i>Chai.A-A</i>	3
<i>Chai.A-E</i>	4

Table 4.4: Number of positive instances extracted for Q1-Q2 context defining features (out of 1522 possible ones).

Feature	Num of found positive features
<i>Center.Ref</i>	104232
<i>Center.Forward</i>	194712

Table 4.5: Number of positive instances extracted for A2-finding features (out of 465732 possible ones).

Transition	Num of found features
<i>Center-one</i>	383
<i>Center-two</i>	10
<i>Center-three</i>	4
<i>Center-four</i>	1125

Table 4.6: Number of instances extracted for each possible Q1-Q2 context defining transition features (out of 1522 possible ones).

One thing that could prove interesting would be to try and test these features on a different dataset. It would be interesting to see how and if the evaluation of the different features would produce different results. Comparing the performance of the features on two different, independent datasets would help shed light on whether the effects seen with Bob data are more general or are specific to this particular dataset and system.

### Stanford Parser

The work described here requires deep syntactic and semantic processing of the data. As such, it requires the use of linguistic tools in order to obtain this information. One such tool that I used is the Stanford parser. In examining the parses of the Bob questions, there were several cases that could be deemed

problematic. That is, there are instances where the parser was not able to properly handle the data. I found two types of such instances.. The first are the instances where the parse produced was somehow incorrect. The other are instances of utterances which are not full grammatical sentences.

```
(37) How can I order books?  
[advmod(i-3, how-1)  
aux(i-3, can-2)  
nn(books-5, order-4)  
dobj(i-3, books-5)]
```

Example 37 is a clear example of a wrong dependency parse being produced. The relation *dobj*, or *direct object* has the noun *I* as its verb argument and the noun *books* as the direct object of *I*. This is obviously wrong. This wrong parse results in the wrong predicate-argument structure being extracted, which in turn can make the feature extraction process erroneous as well.

```
(38) DVDs  
[]
```

Example 38 is a typical example of a non-sentential utterance. Since the parse produced is empty there is no predicate-argument structure that is extracted, which in turn signifies that all the features have a value zero for the snippet that contains this particular question.

More work needs to be put into extracting the predicate-argument structure, as well as ways to handle non-sentential utterances such as the one in example 38.

## Chapter 5

# Conclusion

The work presented here shows how deep and semantic processing can be beneficial to the study and implementation of IQA systems. Moreover, it shows the importance that context holds in IQA, and the ways that this context can be modeled. Most importantly what the work described above shows is that context and deep analysis alone do not result in best the best performance. Rather, it is the combination of different types of processing (both shallow and deep) as well as different type of information that can be extracted (context modeling and answer seeking) that produces the best results. The evaluation above shows how deep analysis and context modeling is beneficial to the performance of IQA systems, but more work needs to be done to expand and improve on this work.

# References

- [1] Raffaella Bernardi and Manuel Kirschner. Context modeling for IQA: The role of tasks and entities. In *KRAQ08, Manchester, UK*, 2008.
- [2] Nria Bertomou, Hans Uszkoreit, Annette Frank, Hans-Ulrich Krieger, and Brigitte Jrg. Contextual phenomena and thematic relations in database QA dialogues: results from wizard-of-oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL*, 2006.
- [3] Bob. BoB, a chatterbot for the university library. <http://web.inf.unibz.it/bob/?popup=yes&language=en>.
- [4] CLEF. Interactive track for the cross-language evaluation forum. <http://nlp.uned.es/iCLEF>.
- [5] Marco De Boni and Suresh Manandhar. Implementing clarification dialogues in open domain question answering. *Natural Language Engineering*, 11(4):343361, 2004.
- [6] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC 2006*, 2006.
- [7] Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual. [http://nlp.stanford.edu/software/dependencies\\_manual.pdf](http://nlp.stanford.edu/software/dependencies_manual.pdf).
- [8] Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008.
- [9] Jonathan Ginzburg. The interactive stance: Meaning for conversation. <http://www.dcs.kcl.ac.uk/staff/ginzburg/tis.pdf>, 2009.
- [10] Barbara Grosz, Aravind Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 2(21):203–225, 1995.



- [11] GuiTAR. A general tool for anaphora resolution. <http://dces.essex.ac.uk/research/nle/GuiTAR/gtarNew.html>.
- [12] Diane Kelly and Jimmy Lin. Overview of the TREC 2006 ciQA task. *ACM SIGIR Forum*, 41(1):107–116, 2007.
- [13] Manuel Kirschner. Building a multi-lingual interactive question-answering system for the library domain. In *Proc. of the 10th Workshop on the Semantics and Pragmatics of Dialogue (Brandial06), Potsdam, Germany, 2006*.
- [14] Manuel Kirschner. The boB IQA system: A domain experts perspective. In *In Proc. of the 11th Workshop on the Semantics and Pragmatics of Dialogue (SemDial07), Rovereto, Italy, 2007*.
- [15] Manuel Kirschner, Raffaella Bernardi, Marco Baroni, and Le Thanh Dinh. Analyzing interactive QA dialogues using logistic regression models. In *Proceedings AI\*IA 09, 2009*.
- [16] OpenEphyra. Ephyra: Question answering system. <http://www.ephyra.info/>.
- [17] Stanford Parser. Stanford typed dependencies manual. <http://nlp.stanford.edu/software/lex-parser.shtml>.
- [18] Carol Peters. *What Happened in CLEF 2007*, pages 1–12. Springer-Verlag Berlin Heidelberg, 2008.
- [19] Massimo Poesio, Rosemary Stevenson, Barbara di Eugenio, and Janet Hitzeman. Centering: A parametric theory and its instantiations. *Computational Linguistics*, 30(3), 2003.
- [20] S. Quarteroni and S. Manandhar. Designing an interactive open-domain question answering system. *Natural Language Engineering*, 15(1):7395, 2008.
- [21] Allaoua Refoufi. A modular architecture for anaphora resolution. *Journal of Computer Science*, 3(4):199–203, 2007.
- [22] Nico Schlaefer, Jeongwoo Ko, Justin Betteridge, Guido Sautter, Manas Pathak, and Eric Nyberg. Semantic extensions of the ephyra QA system for TREC 2007. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC), 2007*.
- [23] Mingyu Sun and Joyce Y. Chai. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems*, 20(6):511–526, 2007.

- [24] TREC. The TREC ciQA task. <http://www.umiacs.umd.edu/~jimmylin/ciqa/>.
- [25] B.W. van Schooten, R. Op den Akker, S. Rosset, O. Galibert, A. Max, and G. Illouz. Follow-up question handling in the IMIX and ritel systems: A comparative study. *Natural Language Engineering*, 15(1):97–118, 2008.
- [26] Douglas Walton. The place of dialogue theory in logic, computer science and communication studies. *Synthese*, 123:327–346, 2000.
- [27] W.A. Woods. *Semantics and Quantification in Natural Language Question Answering*, volume 17, pages 2–86. Academic, 1978.

## Appendix A

# Sample Bob Snippet Data

Q1: do you have books about comics?

A1: The complete stock of the Library can be found in the OPAC. With the OPAC you can search for a subject. There are of course other resources to search for a special subject. Shall I start to explain some things about the OPAC?

Q2: yes, please

A2: The OPAC offers two possibilities to search by topics. Both can be found in the advanced search menu. Searching for subject is the first possibility. And the second one ...

Q1: yes, please

A1: The OPAC offers two possibilities to search by topics. Both can be found in the advanced search menu. Searching for subject is the first possibility. And the second one ...

Q2: and the second one?

A2: ... is to search for a classification. You simply have to enter one or two letters - for example QP for Business Economics - and the click the LIST button next to the search field. You'll get a list of all classifications for this area. By clicking one you'll get all relevant documents. And you can also search for a specific classification. Do you want to know how it works?

Q1: and the second one?

A1: ... is to search for a classification. You simply have to enter one or two letters - for example QP for Business Economics - and the click the LIST button next to the search field. You'll get a list of all classifications for this area. By clicking one you'll get all relevant documents. And you can also search for a specific classification. Do you want to know how it works?

Q2: yes

A2: You only have to insert the complete classification, like ES 681, and you get all relevant documents - in this case about foreign language teaching in general.

Q1: yes

A1: You only have to insert the complete classification, like ES 681, and you get all relevant documents - in this case about foreign language teaching in general.

Q2: and what about dvd?

A2: Finding DVDs is rather easy with the catalogue. Just open the advanced search screen and choose "Film - Video" as type of medium. If you want to narrow your search to films only you can enter "AP" in the classification field.

Q1: and what about dvd?

A1: Finding DVDs is rather easy with the OPAC. Just open the advanced search screen and choose "Film - Video" as type of medium. If you want to narrow your search to films only you can enter "AP" in the classification field.

Q2: and cd-rom?

A2: With the advanced search you can refine your search to show only specific types of media (books, DVDs, journals, etc.)

Q1: and cd-rom?

A1: Some of our databases are only available on CD-ROM. They are accessible from within the University network only. In this database list you can recognize them through the disk icon.

<http://www.unibz.it/library/databases/index.html?LanguageID=EN>

Q2: and online databases?

A2: Well, there are lots of online databases. I'd recommend this page, because it lists all subject specific databases the Library offers.

Q1: and online databases?

A1: Well, there are lots of online databases. I'd recommend this page, because it lists all subject specific databases the Library offers.

<http://www.unibz.it/library/databases/index.html?LanguageID=EN>

Q2: what's refworks?

A2: RefWorks is web-based bibliographic software package that enables you to: Organize your research, include citations while you write your paper, build a bibliography in a variety of formats, import references from many data sources, and create bibliographies in different document formats. On this page you get more information.

Q1: can i study in the library ?

A1: To get a Library Card you have to come to the Library in Bozen-Bolzano or Brixen-Bressanone. You only need an ID and 10 to pay the annual fee. Your personal Library Card will be ready within ten minutes. If you have a Student Card of the University Bozen-Bolzano you can use it as your Library pass.

Q2: when does the library open in the morning ?

A2: You can find our opening hours on this website.

Q1: when does the library open in the morning ?

A1: <http://www.unibz.it/library/openinghours/index.html?LanguageID=EN>

Q2: where is the main building in brixen ?

A2: The Library is located in the main buildings of the University. In Bozen-Bolzano the entrance is on the first floor. In Brixen-Bressanone you enter the Library on the ground floor.

Q1: where is the main building in brixen ?

A1: The Library is located in the main buildings of the University. In Bozen-Bolzano the entrance is on the first floor. In Brixen-Bressanone you enter the Library on the ground floor.

Q2: can i take the books out of the library ?

A2: If you want to borrow books, DVDs or journals you need to take them from the shelf yourself and check them out at the information desk. If you want them to be sent to Brixen-Bressanone, Bozen-Bolzano, Bruneck-Brunico or Schlanders-Silandro you can order them via the OPAC.

Q1: can i take the books out of the library ?

A1: If you order a book from Brixen-Bressanone for Bozen- Bolzano or the other way round it normally takes two days to arrive. If you choose Bruneck-Brunico or Schlanders-Silandro as destination will take a bit longer.

Q2: can I take journals out of the library ?

A2: It's the same like borrowing a book. Just get it from the shelf and check it out at the circulation desk. The only difference is the reduced lending period of 2 weeks.

Q1: can I take journals out of the library ?

A1: The circulation desk is the place to borrow items. Do you want to know more about it?

Q2: no

A2: Ok.

Q1: no

A1: Gosh, I didn't quite get that. Strange, since I am actually pretty intelligent.

Well, I will ponder upon that further and do some research, so the issue will make more sense to me on your next visit.

Q2: you are a pretty smart guy Bob..don't worry

A2: Oh, thanks for the compliment! If I could I would blush now ;-)

## Appendix B

# Stanford Parser

# Dependency Relations

*abbre* - abbreviation modifier  
*acomp* - adjectival complement  
*advcl* - adverbial clause modifier  
*advmod* - adverbial modifier  
*agent* - agent  
*amod* - adjectival modifier  
*appos* - appositional modifier  
*attr* - attributive  
*aux* - auxiliary  
*auxpass* - passive auxiliary  
*cc* - coordination  
*ccomp* - clausal complement  
*complm* - complementizer  
*conj* - conjunct  
*cop* - copula  
*csubj* - clausal subject  
*csubjpass* - clausal passive subject  
*det* - determiner  
*dobj* - direct object  
*expl* - expletive  
*infmod* - infinitival modifier  
*iobj* - indirect object  
*mark* - marker  
*measure* - measure-phrase modifier

*neg* - negation modifier  
*nn* - noun compound modifier  
*nsubj* - nominal subject  
*nsubjpass* - passive nominal subject  
*num* - numeric modifier  
*number* - element of compound number  
*parataxis* - parataxis  
*partmod* - participial modifier  
*pcomp* - prepositional complement  
*pobj* - object of a preposition  
*poss* - possession modifier  
*possessive* - possessive modifier  
*preconj* - preconjunct  
*predet* - predeterminer  
*prep/prepc* - prepositional modifier  
*prt* - phrasal verb particle  
*punct* - punctuation  
*purpcl* - purpose clause modifier  
*quantmod* - quantifier phrase modifier  
*rmod* - relative clause modifier  
*ref* - referent  
*rel* - relative  
*tmod* - temporal modifier  
*xcomp* - open clause complement  
*xsubj* - controlling subject



## Appendix C

# MiniProbBank Sample

```
- <task name="read" stemmed="read">
  <variant stemmed="review">review</variant>
  <variant stemmed="studi">study</variant>
  <relatedTask similarity="0.3">photocopy</relatedTask>
  <relatedTask similarity="0.1">pay</relatedTask>
- <arg0>
  - <entity relType="isa">
    <canonical stemmed="user">user</canonical>
  </entity>
</arg0>
- <arg1>
  - <entity relType="isa">
    <canonical stemmed="item">item</canonical>
  </entity>
</arg1>
- <argm-loc>
  - <entity relType="isa">
    <canonical stemmed="librari">library</canonical>
  </entity>
</argm-loc>
- <argm-mnr>
  - <entity relType="isa">
    <canonical stemmed="card">card</canonical>
  </entity>
</argm-mnr>
- <argm-tmp>
  - <entity relType="isa">
    <canonical stemmed="time">time</canonical>
  </entity>
</argm-tmp>
</task>
```

Figure C.1: An example of a task entry in the MiniPropBankFile.

## Appendix D

# GuiTAR Output Sample

```
-<s id="s68">
  -<sbar id="sbar8">
    <W Lpos="WRB">how</W>
    <W Lpos="JJ">much</W>
    -<unit id="unit18">
      <W Lpos="AUX">is</W>
      -<unit id="unit19">
        -<ne id="ne96" AACat="pers-pro" AAper="per3" AAnum="sing" AAgen="neut">
          -<nphead id="AAh87">
            <W Lpos="PRP">it</W>
            </nphead>
          </ne>
          <W Lpos="IN">per</W>
          -<ne id="ne97" AACat="bare-np" AAper="per3" AAnum="sing" AAgen="neut">
            -<nphead id="AAh88">
              <W Lpos="NN">article</W>
              </nphead>
            </ne>
          </unit>
        </unit>
      </sbar>
    -<AAante current="ne96" rel="ident">
      <anchor antecedent="ne95"/>
    </AAante>
  </s>
```

Figure D.1: An example of the XML node for a given sentence as produced by GuiTAR.