Answering open-domain temporally restricted questions in a multi-lingual context

Rob Basten

24th August 2005

Summary

Question answering (QA) is the part of the information retrieval field, where research is conducted to give an exact answer to a question in natural language instead of a set of documents as response to a query. The field of QA is moving from answering simple questions like *'Who painted the Mona Lisa?'* to answering complex questions like *'Which countries neighbour with Belgium and Germany?'* or *'Where did Stefan Zweig move after the Second World War had begun?'*.

It was clarified that complex questions are questions whose answers need to be gathered from pieces of information divided over different documents. This leads to the three steps in which QA can roughly be split: question analysis, document retrieval and answer construction. The first and last of which need to be adapted in order to answer complex questions. In this thesis, the focus has been on the question analysis.

One type of complex questions was chosen to be handled in this thesis: temporally restricted questions. These are questions like *'Where did Stefan Zweig move after the Second World War had begun?'* and *'Who was queen in 1945?'*. The goal of this thesis has been to answer these questions using a 'normal, existing QA system'. The questions can be posed in German and answers are searched for in German or English. The questions are open-domain, so the questioner can chose any subject he or she likes.

Two different strategies have been worked out and put to work together. The first strategy is to let the existing QA system answer the question and check the correctness of the retrieved answers when an explicit date or period is in the question. The second strategy is to split a question in its 'real question' and its temporal restriction. For the example given above, this would lead to respectively *'Where did Stefan Zweig move?'* and *'When did the Second World War begin?'*. The first strategy would probably lead to a high precision and the second to a higher recall.

The designed system was implemented and tested during CLEF 2005. For the task with German questions and answers, eleven temporal questions were answered correctly. The run was repeated after some bugs had been removed from the initial system and only eight questions were still correctly answered. It was analysed what caused this worse performance and how the system came to the correct answers. Just one answer was only found by using the second strategy, so splitting the questions (two answers were found with both strategies). Still it is thought this second strategy is useful and the somewhat meagre result is mainly due to the CLEF questions being just reformulations of sentences in the corpus the questions should be answered from. Splitting is not useful in those cases. In a 'normal' situation, the questioner does not construct a question by examining the answer corpus.

Preface

This thesis is written as the conclusion of my study of Computer Science at the University of Twente (UT). I have written my thesis from January till August 2005 at the Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), at the 'Language Technology' research lab (LT), which is led by prof. dr. Hans Uszkoreit. At this point I would like to thank him and the DFKI in general for giving me this possibility. At the DFKI, my supervisor has been dr. Günter Neumann. I would like to thank him for the practical support and the comments on and ideas about my thesis. When thanking for practical support at the DFKI, Bogdan Sacaleanu cannot be forgotten. He helped a lot during implementation and testing.

At the UT, the chair of my supervising committee has been prof. dr. Franciska de Jong. She was also responsible for contacting the DFKI in the first place. I would like to thank her for her work she has done even before I was graduating and for her extensive comments on the draft versions of my thesis. As third member of my committee, I would like to thank dr. ir. Rieks op den Akker for his help. Finally, I would like to thank my parents (and brothers) for their support and their moving me from Enschede to Saarbrücken to Middelrode and Manon Penning for her support and comments on early draft versions of this thesis.

Rob Basten

Contents

List of Figures

List of Tables

Definitions and abbreviations

Note: These definitions do not capture all meanings of a word or abbreviation, but only the meaning(s) relevant for this thesis.

anaphora	The use of a linguistic unit, such as a pronoun, to refer back to another unit. For example, 'his' in: 'Bush is president of the US. Clinton was his predecessor'
anaphora resolution	The removal of the referring unit of an anaphora and its substitution by the unit it referred to
CLEF	Cross Language Evaluation Forum, since 2003 it comprises the QA@CLEF track. If not explicitly stated otherwise, just QA@CLEF is meant by CLEF
definition question	Question for which an answer consists of a couple of information nuggets, together defining the subject asked for in the question. In CLEF only persons or organisations, but in general these subjects could be any 'object'. Definition questions are for example: 'Who is Heinrich Böll?' or 'What is a computer?'
dialogue system	System that interacts with its users in order to give them information or accomplish some task for them. Instead of just giving answers like a QA system it can remember the topic of earlier questions and use this knowledge for its understanding of current utterances of the user. It could also ask the user to clarify its utterance
EAT	Expected answer type, the type of answer expected for a given question. For example: 'When did Newton die?' suggests a date as answer
ellipsis	The omission of a word or phrase necessary for a complete syntactical construction but not necessary for understanding. For example: 'Clinton was president before Bush.' Which stands for: 'Clinton was president before Bush was president'
ellipsis resolution	The removal of ellipsis from a sentence in order to form the full sentence it originated from

factoid question	A fact-based question for which an answer consists of only one or a few words, often named entities, like the name of a person, location or organisation. Other possibilities for example are a number or a date. The answer can mostly be found in one line in one document. For example: 'Who painted the Mona Lisa?'
list question	Question for which an answer consists of a list of answers, mostly factoid answers. For example: 'Which countries rejected the European Union's first constitution?'
main clause	The main part of a sentence that consists of two clauses. For example the part until the comma in: 'Where did Clinton live, before he became president?'
monolingual	In the context of QA systems: questions, answers and corpus from which the answers are retrieved, are in the same language
multilingual	In the context of QA systems: questions, answers and corpus from which the answers are retrieved, are not all in the same language
multimodal	In the context of language technology: using more than one way to communicate, like text, speech, pictures etc.
NE	Named entity, an entity in a sentence with a name; a person, location or date for example
NE annotator	System that looks for named entities in a sentence and tags them
NL	Natural language, the language people use, as opposed to language computers use
NLP	Natural language processing
NP	Noun phrase, a part of a sentence that consists at least of a noun and possibly of multiple nouns, adjectives and/or a determiner
open-domain	Questions posed are not restricted to one or a couple of subject area(s), but instead, questions can ask for anything.
POS	Part-of-speech, a word is a determiner, adjective, noun or verb for example. This is called its part-of-speech
POS-tagger	System that assigns a (or multiple) part-of-speech to each word in a sentence

PP	Prepositional phrase, a part of a sentence that consists at least of a preposition, which is the head of the PP, and an NP. It could possibly contain another PP
precision	Let: ω be a randomly chosen answer from all possible answers $y = \begin{cases} 1, \text{ if } \omega \text{ is a correct answer for a posed question} \\ 0, \text{ otherwise} \end{cases}$ $y' = \begin{cases} 1, \text{ if the algorithm detects } \omega \text{ for a posed question} \\ 0, \text{ otherwise} \end{cases}$ Then: recall $= P(y' = 1 \mid y = 1)$, precision $= P(y = 1 \mid y' = 1)$
QA	Question Answering, the automatic answering of questions, not returning complete documents, but only short answers as a human would do
QA@CLEF	The QA track or part of CLEF
recall	See explanation under 'precision'
snippet	A piece of text, extracted from a document or website. For example, Google returns a couple of titles, with for each title a snippet of its website and its URI
subordinate clause	The subordinate part of a sentence that consists of two clauses, mostly starting with a subordinating conjunctive. For example the part starting after the comma (with conjunctive 'before') in: 'Where did Clinton live, before he became president?'
template	Templates are used (in QA) for multiple purposes: • For the answering of definition questions. For example:When a definition of a writer is asked for, a template is used to check which information should be found and returned, in this case among others date of birth and death and his main books • For the generating of an answer sentence (when not just an answer is returned, but instead a complete sentence). For example: When the question asked for someone's date of birth, the template could be 'He was born on ANSWER'
TREC	Text REtrieval Conference, since 1999 it comprises a QA track

1 Introduction, research questions and approach

Information Retrieval (IR) is a broad field of research where attempts are made to automatically extract information (mostly documents) a user is looking for. A well known example of IR are the search engines on the web, like Google, in which a user can give a query like 'MOVED "STEFAN ZWEIG" "SECOND WORLD WAR"' in the hope documents are returned where the user can find that Stefan Zweig moved to Brazil after the Second World War had begun. What the user probably would like better is to have the possibility of asking a system *'Where did Stefan Zweig move after the Second World War had begun?'* and having the system return *Brazil*. In the field of Question Answering (QA), which is part of the IR field, research is conducted to make this possible.

This thesis has been written at the 'Language Technology' research lab (LT) which is part of the Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI). The DFKI is one of the largest nonprofit contract research institutes in the field of innovative software technology based on Artificial Intelligence (AI) methods. It is located in Kaiserslautern and Saarbrücken, both in Germany. The LT lab is in Saarbrücken.

At the LT lab there is an ongoing research project in the field of automatic question answering, which is called 'quetal'. The question answering system (QA system) that is being built in the context of this project (the system is called Quantico[1]) is used to participate in the QA track of the Cross Language Evaluation Forum (CLEF or QA@CLEF[2]), in 2005 for the third time. QA@CLEF is an international 'contest' where research groups can evaluate their QA systems on a standardized set of questions[3].

1.1 Problem definition

The QA system that was in use at the DFKI could answer 'simple' factoid questions like:

- *When was Mozart Born?*
- *Where is the DFKI located?*
- *Which painter painted the Mona Lisa?*

However, more complex questions could not be answered. Questions like:

- *Which countries neighbour with Belgium and Germany?*
- *Where did Stefan Zweig move after the Second World War had begun?*

[1] See Neumann and Sacaleanu (2005) on the working of the Quantico system.

[2] CLEF is the name of the Forum and QA@CLEF of its QA track. In this thesis however, if not explicitly stated otherwise, CLEF is used to denote just the QA track of CLEF.

[3] More information on CLEF can be found in Section 2.2.

- *How should I assemble a bike?*

The first problem here was that although everyone probably 'feels' there is a difference between the first three and last three questions, it is difficult to say what this difference exactly is. This has led to the first research question being: What are complex questions? The research questions are looked at in the next section.

Answering a question in QA can roughly be split into three steps:

1. Question analysis: Analysing the question posed by the user and constructing a query that can be used for the retrieval of the information units. For example: constructing something like 'MOVED "STEFAN ZWEIG" "SECOND WORLD WAR"' from *'Where did Stefan Zweig move after the Second World War had begun?'*.
2. Document retrieval: Retrieval of 'relevant' passages or documents. For example a passage like *'Zweig then lived in England (in Bath and London), before moving to the USA. In 1941 he went to Brazil, where he and his second wife Lotte (née Charlotte E. Altmann) committed suicide'*.
3. Answer construction: Deciding which information element or elements should be extracted from the 'relevant' passages and should be returned to the questioner. *Brazil* in the given example.

The focus in this thesis is on the *question analysis* of *complex questions* and the possibilities of using an existing QA system that can already answer 'simple' questions (the Quantico system) to answer these complex questions. The system was meant to be implemented so it could really be used (for CLEF).

The existing QA system, Quantico, and therefore the QA system which is the focus of this thesis (a QA system that can answer complex questions), are further restricted in that:

- The QA system works on an *open-domain*, so the question can be concerned with all kinds of subjects. This opposite to systems working on a closed-domain and for example answering only questions about the solar system.
- The first part of a question, so *'In which country'* in *'In which country is the DFKI located?'* needs to be posed in *correct* natural language. In this case the sentence can be parsed. When the total question is not a well-formed sentence (grammatically correct) and without spelling errors, a useful parse is not probable.
- Questions are posed in *German* and answered in *German* or *English*.

  - Also English questions are handled (and answered in German), but they are first translated and only then analysed and answered, which causes them to be effectively German questions for the system.
  - Dutch questions with German answers were originally also planned. The main problem with this combination was the unavailability of a parser for Dutch that could be easily connected to the existing system. Therefore, it would take a lot of time to get this language-combination to work and it was decided not to implement this combination.

- Just one question is handled at a time, so there is no context knowledge and the questioner can not be asked to explain his or her question.
- Answers are derived from *unstructured data*: a newspaper corpus. (Sometimes, it is argued a newspaper corpus is not really unstructured, like for example the Internet is. Newspaper corpuses contain (as good as) only correct sentences, as opposed to the Internet. Newspaper corpsuses are then called *semi-structured data*.)
- Answers are not generated text, but only *extracted snippets*, often just one or two words.
- For CLEF, only one answer may be returned, instead of a list of possible answers.

1.2 Research questions

Because it was not clear what complex questions are, it was decided this should be clarified, so that one type of complex questions could be chosen to cope with in this thesis. The research questions that arised were:

1. What are complex questions?
   - Do different views exist about what complex questions are in the context of open-domain QA?
2. Which type(s) of complex questions is interesting to handle?
   - Which type(s) can function as a prototype for other complex types? (So that other types of complex questions can be dealt with in a similar way.)
   - Are there other reasons to choose a type?

After these questions were answered questions with a temporal restriction were chosen as the type of complex questions to handle in this thesis. Temporally restricted questions are for example:

- *Who was queen of the Netherlands in 1900?*
- *Where did Stefan Zweig move after the Second World War had begun?*

At that point, the main research question could be formulated:

> How can, in the context of open-domain multi-lingual question answering, temporally restricted questions be answered using an existing question answering system that was designed to answer only 'simple' factoid questions?

The research questions that arised from this main research question are:

3 What is the state-of-the-art in handling complex questions (with focus on the temporally restricted questions)?
   - Are there research groups that already handle the questions?
   - How do these groups handle them?

4 How could the chosen type of complex questions be handled?
   - What are the options?
   - Which options seem to be hopeful and need further attention?
   - Which option performs best?

5 How can the gained knowledge be used to improve the DFKI QA system that is applied at CLEF 2005 and coming QA@CLEF tracks?

1.3 Approach

The first research question, 'What are complex questions?', was answered by studying the papers written in the context of the CLEF and TREC (Text REtrieval Conference) QA tracks. A few queries with Google have led to some other papers and through them to the proceedings of the yearly ACL (Association for Computational Linguistics) and NAACL (North American chapter of the Association for Computational Linguistics) meetings.

Then, temporal questions have been selected as the complex questions to deal with (second research question). The state-of-the-art in handling complex questions, focused on the temporal questions, (the third research question) was easily answered, since the papers that mention complex questions are usually also the papers in which the handling of complex questions is further elaborated on. However, complex questions are not dealt with a lot.

After that, an initial way of handling the chosen complex questions, temporal questions, was chosen (the fourth research question). An initial system was built and used for QA@CLEF 2005 (fifth question). The performance of this system has been evaluated and suggestions could be made for future progress.

1.4 Arrangement of the next chapters

In Chapter 2, the history and future of QA, as expected by experts in the field, is shown in order to clarify how QA has moved to where it is now, thereby giving a better understanding of why complex questions are interesting to handle at this point in time. In Chapter 3 it is explained what complex questions are and what causes questions to be complex. At the end of this chapter it is explained which complex questions have been chosen to further elaborate on in this thesis and why.

In Chapter 4, the state-of-the-art in complex questions is sketched, focusing on the temporal questions. After that, in Chapter 5, it is explained what ways there were in the context of this thesis to deal with temporal questions and it is explained which way was chosen.

Chapter 6 describes the implementation of the chosen system and the results of the evaluation can be found in Chapter 7. These results also comprise the results of the 2005 QA@CLEF. This thesis ends with Chapter 8, containing the conclusions and recommendations.

2 History, present and future of QA

In this chapter, the first research question is answered: 'What are complex questions?' In the following section of this chapter a brief history of QA systems can be found that explains how the focus in today's research has moved to complex questions on an open-domain. In the second section, the forces in the current field of QA are worked out, followed, in section 2.3, by the state-of-the-art in QA systems and the problems that have to be tackled in the near future.

2.1 Overview of previous QA systems

Since the sixties and seventies QA systems have been built. Winograd (1972) built one in the seventies SHRDLU, that represented a robot that could answer questions and execute commands from a user that related to a number of coloured objects in a small world. The system can be seen as a natural language interface to an expert system (Prager et al., 2004). In his book, Winograd (1972) mentions four different kinds of systems that were built earlier:

1. Systems like BASEBALL (P. Green et al., 1963) and ELIZA (Weizenbaum, 1966), 'special format systems' as Winograd calls them, used one format to store the knowledge in a database and used another format to extract meaning from the English inputs (the questions). They returned short answers in natural language and got good results in small areas on factoid questions[4]. However, because information had to be stored in a specific format, they were very inflexible in that they could not be used in another domain than the one they were designed for. The main challenge for these systems was the transformation of natural language questions into a database query (Monz, 2003).

2. Another type of QA systems were the 'text based systems' that stored text which was indexed. When a question was given, the words in the question should match with the indexed words, or nothing would be found. Complete sentences were retrieved. Flexibility in these systems was low as words should match exactly those in the indexed text. An example is the PROTOSYNTHEX 1 by Simmons (1966).

3. The third type were the 'limited logic systems', in which a sort of formal notation is substituted for the English sentences in the database. These systems could handle somewhat more complex questions, where complex means "... questions which are not answerable by giving out

[4]See definitions.

one of the database assertions, but demand some logical inference to produce an answer" (Winograd, 1972, p. 37). The complex information was not part of the data, but was built into the system programs. Some systems for example could use simple logical relations like 'if A is a part of B and B is a part of C, then A is a part of C', but this relationship had to be stored in the program, so that they could only handle relationships they were designed for. Moreover they could only accept simple assertions as input, so not a question like *'Is there a country which is smaller than every U.S. State?'*. Two examples of these systems are SIR (Raphael, 1968) and DEACON (Thompson, 1966).

4. The problems of limited logic systems, as mentioned at the end of the former bullet, were recognized very early (Winograd mentions Raphael, 1968, p. 90). The 'general deductive systems', the fourth type of systems, used some standard mathematical notation to store knowledge, for example predicate calculus. The Robinson resolution algorithm (Robinson, 1965) was a very simple uniform proof procedure for first-order predicate calculus that was (1) uniform, so the system did not have to, and even could not, be told how to go along proving things, and (2) if a proof consisted the algorithm was sure to find it, although this could take a long time. An example of these systems was QA3, built by C. Green and Raphael (1968).

   The 'procedural deductive systems', that Winograd seems to treat as part of the 'general deductive systems' tried to overcome the problem that it sometimes took very long to find an answer when using the Robinson resolution algorithm. Procedural information was used to get faster to an answer. Care was taken not to store information in a way it could be used only for one purpose. An example of such a system was built by Woods (1968) and is called LUNAR. Although this system was relatively advanced, it could still be best described as a natural language interface to a database (Prager et al., 2004).

Disadvantages of these early systems, according to Krause (1982), were (1) they could not handle large grammars because of insufficient system resources. Therefore, choices had to be made as to which rules would be in the grammars. In this way, it was often the case that instead of a couple of different possibilities of asking a question, a user could only use one way of asking it. (2) The domain the systems could answer questions from was small.

Until the early nineties, there have been few further research efforts in this area. This is noticed for example by Monz (2003)[5] and Diekema et al. (2003), but can also be seen in that practically never other systems are referenced than those from the sixties and early seventies. Of course, there have been some efforts, like the CHAT-80 system built by Warren and Pereira (1982), but in the beginning of the nineties, Lang et al. (1990) note that quality of computer-generated answers has been very unimpressive in expert systems and intelligent decision support systems.

2.2 Current initiatives in the field of QA

Recently a couple of programs have given a major impulse to the field of QA. Of these programs, the Text REtrieval Conference (TREC) is a very important one (see for example Diekema et al., 2003; Manning, 2004; Monz, 2003; Tanev et al., 2004). TREC is a program meant to encourage research in information retrieval of large text collections. It introduced a QA 'track' in 1999 (TREC 8). As already mentioned in Chapter 1, QA can be seen as a part of the information retrieval field. "The goal of the track is to foster research on systems that retrieve answers rather than documents in response to a question, with particular emphasis on systems that can function in unrestricted domains" (Voorhees, 2003, p. 54)[6]. The task in the first two QA tracks (TRECs 8 and 9) was the same: For each question, systems should retrieve a ranked list of up to five text snippets that contained an answer to the question plus a document that supported the answer. The questions were restricted to factoid questions, so questions like:

- *Who painted the Mona Lisa?*
- *Where was Kohl born?*

The document collection from which the answers should be extracted were standardized and given to the participants. This is generally done in programs like TREC.

In TREC 2001 (also known as TREC 10) two new tasks were introduced:

- The list task, containing list questions, so questions for which a list of answers has to be returned. For example a question like:
  - *Which countries border Germany?*
- The context task. "The context task was a pilot evaluation for question answering within a particular scenario or context. The task was designed to represent the kind of dialogue processing that a system would

[5] In his book, Monz has an overview of previously built QA systems, partly based on Simmons (1965, 1969), that is very suitable for further reading on this subject.

[6] For information on the QA tracks of TREC until 2004 see Voorhees (2002, 2003, 2004).

> need to support an interactive user session. Questions were grouped into different series, and the QA system was expected to track the discourse objects across the individual questions of a series" (Voorhees, 2002, p. 58).

Also, in the main tasks (the factoid questions), the guarantee that an answer to every question was in the document collection was eliminated. Instead, *NIL* had to be returned if the system believed an answer could not be found.

TREC 2002 contained two tasks that were already in TREC 2001, the main task and the list task, but systems were required to return exact answers. In TREC 2003 the definition questions were introduced, so questions like:

- *Who was Theo van Gogh?*
- *What is mold?*

The factoid, list and definition questions were tagged as to its type.

In 2004 the questions were arranged in groups that had a target like 'Fred Durst'. In every group were a couple of factoid and list questions and at the end one question that simply stated 'other'. The 'other' question was a kind of definition question, but information already in the former answers of the group should not be in the answer for the 'other' question. At the moment of writing, it has not yet been announced in which way the QA track has changed for 2005 (it is announced the QA track continues in 2005).

The Cross-Language Evaluation Forum (CLEF) is meant to support research in cross-lingual information retrieval. Since 2003 it contains the QA@CLEF track, which can be compared with the QA track at TREC. As Magnini et al. (2004)[7] put it: "...the guidelines reflected to a large extent those of the TREC 2002 QA track, adopting similar requirements and evaluation measures" (p. 282) except that CLEF focuses on "...challenging tasks in languages other than English or even across different languages..." (p. 281).

In 2003 200 simple, fact-based questions were given as input and the participants had to return up to three responses per question, either exact or 50 bytes-long answer-strings. In 2004, again there were 200 questions per task (combination of source and target language). The questions were fact-based, but about 10% of the questions was made up of definition questions and another 10% did not have an answer in the corpora and *NIL* had to be returned. Also, only one exact answer should be given per question. It was explicitly stated that the definition questions would be about persons and organisations. Therefore, it was relatively easy to handle the definition questions with pattern recognition using a number of templates or 'linguistic regular expressions' (see for example Pérez-Coutiño et al., 2004; Tanev et al., 2004).

[7]For information on the QA@CLEF see this paper (Magnini et al., 2004).

In 2005 nothing has changed to the definition and factoid questions in CLEF. However, there was a third task, namely the 'temporally restricted factoids' (questions normally known as factoids were called 'temporally unrestricted factoids'). These 'temporal questions' were restricted by:

1. Date, for instance *'Who was Russia's president in 1999?'*.
2. Period, for instance *'Who was Russia's president between 1999 and 2001?'*.
3. Event, for instance *'Who was Russia's president when Chodorkovski was taken prisoner?'*.

This same task had been in the 2004 QA@CLEF as pilot for Spanish (see CLEF, 2004).

A couple of other programs that, among other things, try to push forward research in QA systems are mentioned in the 'Vision Statement to Guide Research in Question & Answering (Q&A) and Text Summarization' by Carbonell et al. (2000) and the 'Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A)' by Burger et al. (2000)[8], but are not discussed here, because they are of lesser importance. Also a number of (annual) congresses exist, for example the annual meetings of the Association for Computational Linguistics (ACL)[9] and its North American chapter of the Association for Computational Linguistics (NAACL). Clearly, the field of QA is moving forward, steered by a number of initiatives.

2.3 State-of-the-art and future of QA systems

At the end of the last section two papers were mentioned by Burger et al. (2000) and Carbonell et al. (2000). These are meant to provide a background against which natural language processing (NLP) research sponsored by the Defense Advanced Research Project Agency (DARPA), the Advanced Research and Development Activity (ARDA) and other agencies can be conceived and guided. Therefore the focus in these papers is mainly on intelligence agencies using QA systems, but they are still very interesting in the context of this thesis: the two papers are written by a large number of (well known) researchers from the QA field. Besides, the aforementioned agencies sponsor for example TREC in general and the QA track in particular, so do not limit their focus to QA in intelligence agencies.

In Carbonell et al. (2000) four different users of QA systems are distinguished on a spectrum ranging from the TREC 8 QA type questioner to the

[8]The latter paper (by Burger et al., 2000) is made by the QA roadmap committee as a response to the former paper (by Carbonell et al., 2000) to outline a potential research and development path that has as its goal achieving the ideas laid out in this former paper.

[9]See www.aclweb.org, last checked on April 15, 2005.

knowledgeable, intense, high-end professional information analyst. The four users, with the names given by Carbonell et al., are:

- The *casual questioner* asks simple, factual questions which could be answered in a single short phrase. For example:
  - *'Where is the Taj Mahal?'*

  This is the 'TREC 8 QA type questioner' and these types of questions have also been in the QA@CLEF track since its start. In TREC 2004, 63 runs were submitted from 28 participants. The best run of the best performing system scored an accuracy of 83.9% on factoid questions[10], meaning 83.9% of the questions were answered correctly. Only the three best performing groups had one (or more) run(s) with an accuracy above 50%. The best run of the sevent-best performing group already had an accuracy below 30%.

- The *template questioner* asks questions that can be answered by creating 'standard templates'. This means that for each question of one type the same template is used with open slots that should be filled for the specific question. The information needed to fill these slots will be found in a number of different documents. Examples of these questions are:
  - *'What is the biography of Jan Peter Balkenende?'*
  - *'What do we know about the DFKI?'*

  For a biography a template could be used like '<NAME> *was born in* PLACE OF BIRTH *on* <DATE OF BIRTH>...'.

  Another possibility at this level are questions that result in producing a list of similar items. Problems here are that it is not known when all the answers are found, that overlapping results which are differently identified should be removed and that every item in the list might include multiple entries (second example). Examples are:
  - *'What are the products being marketed by Sun?'*
  - *'Name all states in the USA and their capitals.'*

  The two types of questions sketched above are (1) the 'definition' (in TREC 2003 and CLEF 2004 and 2005) or 'other' (in TREC 2004) questions and (2) the 'list' questions (in TREC 2001 – 2004).

  For list questions, the distinct answers returned for each question are important. For example a question like the second one given above has fifty distinct correct possible answers. Precision is the number of

[10]Because in TREC 2004 questions were ordered in groups with a common subject, a distinction was made between 'initial' and 'non-initial' factoid questions. The initial question of a group is often more fully specified than later questions. Therefore the results on the initial questions are used in this thesis.

correct distinct responses divided by the number of returned responses (if a list of 30 answers is returned, of which 24 are correct, precision is 80%). Recall is number of correct distinct responses divided by the number of possible correct responses (for the question asking for all the states in the USA, when 24 correct answers are in the list, the recall is 48%). For the list questions in TREC 2004 only the F-measure is given in Voorhees (2004). The F-measure combines with equal weight precision and recall (when $\beta = 1$ is used) to form one score: $F = \frac{2 \cdot P \cdot R}{P+R}$, where P is precision and R is recall. The F-score for the best run of the three best performing groups were respectively 62.2%, 48.6% and 25.8%.

For the 'other' questions in TREC 2004 'correct nuggets' are defined for each question. So for a question with topic *Clinton*, all the information nuggets are correct that an assessor has found to be important, for example his date of birth, the date he became president etc. Recall is now defined analogously to the list questions. So when a topic like *Clinton* has ten relevant nuggets and the system extracts six correct ones, recall is $\frac{6}{10} = 60\%$. "Nugget precision is much more difficult to compute since there is no effective way of enumerating all the concepts in a response." (Voorhees, 2004, p. 6). Therefore, the answer may use 100 characters per correct nugget. So for the example given, 600 characters may be used. In that case precision is 100%. If more characters are used, precision is $1 - \frac{length - allowance}{length}$. The F-measure used for the other questions makes recall three times more important than precision and is defined as: $F(\beta = 3) = \frac{10 \cdot P \cdot R}{9 \cdot P + R}$, where P is precision and R is recall. The F-score for the best run of the best performing group was 46%. Only the five best performing groups scored above 30%.

- The *cub reporter* can be a user that is a 'cub reporter' sent to a city where an earthquake has been. He has to write an article about one aspect of the disaster but needs information, including background information. The QA system should decide or interact with the reporter to know which information should be returned: How deep the system should dig for information and how broad the search should be. Multiple media can be involved and sources in multiple languages, also there could be conflicting facts like the number of citizens killed.

  Interaction with the user is dealt with, for example, in the 'Interactive Cross-Language Information Retrieval' (iCLEF)[11] and searching in image collections in the 'Cross-Language Retrieval in Image Collections' (ImageCLEF)[12], in video in the 'TREC Video Retrieval Evaluation'

[11] See http://nlp.uned.es/iCLEF/, last checked on April 24, 2005.
[12] See http://ir.shef.ac.uk/imageclef/, last checked on April 24, 2005.

(TRECVID)[13] and in sound in the 'Cross-Language Spoken Document Retrieval' (CL-SR)[14]. Multiple languages are covered in CLEF in general and thus also in QA@CLEF. Deciding which information should be returned and dealing with conflicting facts is always a major concern in Information Retrieval (IR) and QA. Dealing with conflicting facts are for example Prager et al. (2004), who use QA-by-Dossier-with-Constraints to find out that, although a system returns *2000* as most likely answer to the question *'When did Leonardo da Vinci paint the Mona Lisa?'*, the right answer should be *1503*, which is the fourth most likely answer. Their system asks when Leonardo da Vinci was born and has died and their system knows a painter is first born, then paints and then dies. By using the answers for the two extra questions, it finds out *1503* is more likely than *2000*.

- The *professional information analyst* can be an investigative reporter, an FBI agent, a historian/writer, a stock broker/analyst, a scientist/-researcher or an intelligence analyst of the CIA. Throughout the document, the intelligence analyst has been selected as the exemplar of this class of users. This class of questioners actually is not very different from the that of the cub reporter. Differences are that the system should have a record of the knowledge the questioner already has, instead of giving all background information. Furthermore, the system should give 'possibilities' when facts conflict or seem inaccurate. Also, the questioner should have the possibility to, for example, give a piece of a broadcast on the television as input to ask the system for information about the unknown military who seems to have influence on the leader of country X.

  Having a record of the knowledge the questioner already has and using it to avoid giving redundant information is partly done in TREC since 2004, in that the answer to the 'other' question should comprise all the interesting information on the topic not already given. Giving 'possibilities' to conflicting facts, has also been part of TREC and CLEF (sometimes by requesting the given answers to be in order of decreasing confidence, sometimes by explicitly asking for a confidence score).

Recapitulating, we could say current research is done mainly to reach to the third stage of the aforementioned four (that of the cub reporter), while still improving results on the first two. Although answering questions posed by 'casual questioners' is not functioning perfectly and answering questions posed by 'template questioners' is also functioning far from perfectly (see

[13] See http://www.itl.nist.gov/iaui/894.02/projects/trecvid/, last checked on April 24, 2005.

[14] See http://clef-clsr.umiacs.umd.edu, last checked on April 24, 2005.

the results for the various tracks and their respective websites), the problems arising when trying to answer questions posed by a 'cub reporter' are now being countered, as can be seen, for example, in that TREC and CLEF are both starting to move their focus towards these types of questions.

3 Complexity of future questions

In this chapter complexity of future questions is elaborated on. In the first section, it can be seen what complex questions are according to the 'Vision Statement to Guide Research in Question & Answering (Q&A) and Text Summarization' by Carbonell et al. (2000).[15] In its subsections each of the characteristics mentioned in the paper is elaborated on.

In Section 3.2 it is summarized what complex questions are and at the end of this chapter, in Section 3.3, a choice for the type of complex question that is dealt with in this thesis, is defended. This is the answer to the second research question.

3.1 Carbonell et al.

Parallel to the shift from the 'casual questioner' to the 'professional information analyst' as seen in Section 2.3, the complexity of questions and answers, according to Carbonell et al. (2000; and also according to Burger et al., 2000), ranges as depicted in Table 1. This table is based on an unnumbered figure in Carbonell et al. (2000, p. 7).

From:		To:	
Questions:	Simple facts	Questions:	Complex; Uses judgement terms; Knowledge of user context needed; Broad scope
Answers:	Simple answers found in single document	Answers:	Search multiple sources (in multiple media/languages); Fusion of information; Resolution of conflicting data; Multiple alternatives; Adding interpretation; Drawing conclusions

Table 1 Complexity of questions & answers (Carbonell et al., 2000)

Complexity of questions, according to the Carbonell et al., can originate from the question side involving:

- Context
- Judgement
- Scope

Complexity of questions can also originate from the answer side when it is needed to:

[15] Why this paper (Carbonell et al., 2000) is a good starting point in this thesis is explained at the start of Section 2.3.

- Use multiple sources
- Use multiple facts
- Interpret retrieved information

In the next subsections, each of these six 'characteristics' is further elaborated on. At first it is described in detail what is meant with the characteristic and then it is shown to which extent researchers agree on complexity originating from the characteristic.

3.1.1 Context

The need for context knowledge is characteristic for the question side of complex questions (Manning, 2004, calls it 'domain knowledge'). For example: Normally the answer to the question *'Where is the Taj Mahal?'* would be *'Agra, India'*, but in another context it could be *'Atlantic City, NJ'*, where the Trump casino, Taj Mahal, is located. Another example is given in Burger et al. (2000; copied from Lehnert, 1978): *'Why did John roller-skate to McDonald's last night?'* would normally mean *'Why did John roller-skate instead of walk or drive or use some other reasonable means of transportation?'*. However, when you know that John is an eccentric health-food nut who roller-skates everywhere he goes, the question focus is on John being at McDonald's instead of John roller-skating.

Graesser et al. (1992) agree on this. They describe three components a QA system should have, one of which is 'recognition of user's goals and knowledge'. Because the QA facility would be extremely intelligent in order to identify goals and knowledge of individual users (references are given to Allen, 1983b; Wenger, 1987), Graesser et al. assume it would be more practical to analyze the typical goals and knowledge of generic users (or N different categories of generic users). In terms of Carbonell et al., this could be the last step before really moving on to the last stage, that of the 'professional information analyst'.

The need for context knowledge is also mentioned by Diekema et al. (2003), who have chosen a "... question-answering strategy that is quite unlike the direction on which TREC has focused" (p. 87). They have developed a QA system that is used in courses for undergraduate students who can ask the system questions concerning 'aeronautical engineering', such as *'What are the changes made to the design of the Shuttle SRM since the Challenger Accident?'*. The system provides five short answers on the answer page. The questions here are quite different from TREC questions in that, among other things:

- The question can be ambiguous
- It can be dependent upon implicit knowledge that is not explicitly stated

This ambiguity of questions and the implicit knowledge relate closely to the 'context' in which a question is posed. As an example they give the question *'Do welding sites yield any structural weaknesses that could be a threat for failure?'* that has no subject. Humans could assume the welding sites are located on the space shuttle. A QA system should also do this.

3.1.2 Judgement

Complex question have a need for judgement when deciding how the question should be translated in multiple questions so that the appropriate relevant information will be retrieved in order to be able to adequately resolve the terms concerning judgement, intent, motive, etc. found in the original question.

Harabagiu et al. (2004) do not explicitly define complex questions, but they give good examples of the need for judgement. They assume that users who ask complex questions associate with it a set of intended questions, that could be factoid or definition questions. Each intended question can generate implied questions. The example they give is, that the question *'What kind of assistance has North Korea received from the USSR/Russia for its missile program?' (Q1)* has (among others) the intended question *'What is the USSR/Russia?'* that has (among others) the implied question *'Is the USSR/Russia the Soviet/Russian government?'*. "When intended questions are generated, their sequential processing (a) represents a decomposition of the complex question and (b) generates a scenario for finding information; thus questions like *Q1* are also known as *scenario questions*"(Harabagiu et al., 2004, p. 31). Implicitly, complex questions are defined as questions that can be decomposed in a number of different questions and judgement can be needed to be able to do this. (This is also related to the 'context' seen in the subsection before and the 'scope' in the next subsection.) Manning (2004) agrees on the need for decomposition of complex questions into a set of simpler questions.

3.1.3 Scope

The last characteristic that can make a question complex on the question side occurs it is not known how broadly or narrowly the question should be interpreted.

Graesser et al. (1992) combine 'judgement' and 'scope' in that they mention 'identification of relevant question-generation mechanisms' as one of three components a QA system should have. Identifying what is relevant, clearly has to do with the scope. Diekema et al. (2003) mention that the question *'How does the shuttle fly?'* is so broad, that it is unknown what the student

exactly wants to know. Furthermore, it can be argued that when a question is decomposed into multiple questions, as seen in the last subsection, the scope is always important. 'Scope' and 'judgement' are closely connected.

3.1.4 Multiple sources

On the answer side, questions are complex because in order to get to an answer, information from multiple, heterogeneous, multi-media, multi-lingual, distributed sources should be retrieved. This is closely linked to the characteristic 'judgement' on the question side. When a question is decomposed into multiple questions, the answers can be extracted from multiple sources.

Diekema et al. (2003) have two examples of this:

1. The question *'At what temperature do liquid metals typically exist?'* is hard, because the answer on this question depends on the kind of liquid metal. The answer therefore has to be searched for in a number of documents or different paragraphs in one document.[16]

2. The question *'What advantages/disadvantages does an Aluminum alloy have over Ti alloy as the core for a Honeycomb design?'* can probably not be answered directly by a QA system. Instead, it should return two documents: The first on the strengths and weaknesses of 'Aluminum alloy' for Honeycomb design and the other on the strengths and weaknesses of 'Ti alloy' for Honeycomb design.

Harabagiu et al. (2004) have a somewhat different view. According to them, list questions and definition questions are not part of the so called 'scenario questions' (complex questions, see Subsection 3.1.2) and they are explicitly not recognized as complex questions. Still it is acknowledged that they are not as easy to answer as factoid questions. This is because "... list questions have answers that are typically assembled from different documents. Such questions are harder to answer than factoid questions because the systems must detect duplications" (Harabagiu et al., 2004, p. 33). So according to Harabagiu et al. the fact that answers have to be extracted from multiple sources doesn't make the question necessarily complex.

Using heterogeneous, multi-media, multi-lingual or distributed sources is not mentioned. However, as seen in Section 2.3, in programs like for example ImageCLEF, TRECVID and CL-SR, searching in multi-media sources is done and in CLEF, multi-lingual QA is done.

[16] First in a number of documents answers to *'At what temperature do liquid metals exist?'* are found. From these answers the correct answer is derived by applying 'typically'. How this should be done exactly is not important here. What is important to recognize is the use of multiple sources (and the decomposition of the question).

3.1.5 Multiple facts

For complex questions, it is often necessary to fuse, combine and summarize smaller facts into larger informational units. Possibly conflicting, duplicate or incorrect facts will be uncovered, which may result in the need to develop multiple possible alternatives with each its own level of confidence. Also, some factual information may degrade over time and that factor would need to be captured in the final answer.

In the last subsection for example, it can be seen that Harabagiu et al. (2004) mention this characteristic and Saquete et al. (2004b) state that complex questions are "... questions whose answer needs to be gathered from pieces of factual information that is scattered in a document or through different documents" (p. 567).

3.1.6 Interpretation

A question is complex, when on the answer side there is the need to interpret the retrieved information appropriately so that the answering component can deal with the terms concerning judgement, intent, motive etc. found in the original question.

According to Graesser et al. (1992), the third component a QA system should have is the 'formulation of an answer based on the first two components, in addition to the knowledge base'. The first two components were mentioned in Subsections 3.1.1 and 3.1.3 and are respectively domain knowledge and generation of relevant questions. This means Graesser et al. think a QA system should interpret the retrieved information and formulate an answer from it. Manning (2004) mentions that answer selection becomes more complicated, since inference based on the semantics of the answer type needs to be activated.

3.2 Recapitulation

In the last section, it can be seen that most authors agree on the characteristics of complex questions mentioned by Carbonell et al. (2000) and Burger et al. (2000). Only the extraction from information from other sources than text, seems to be a bit futuristic for most authors, meaning they do not see this as a characteristic of complex questions in the near future.

Diekema et al. (2003) mention another possible characteristic of complex questions, that is not mentioned by Carbonell et al.. The system they built to answer questions from students related to 'aeronautical engineering' receives questions that can rife with malformed syntax and spelling errors. This causes questions to be harder to answer and therefore to be more complex.

The robustness of question analysis is of importance here (for example using statistics or thesauri to fix spelling errors) and sometimes malformed syntax and spelling errors can be partly fixed with knowledge of the 'context'.

However, the main characteristic of a complex question seems to be, as Saquete et al. (2004b, p. 568) state it, that they "... have in common the necessity of an additional processing in order to be solved", so questions have to be split up in multiple questions and multiple facts, retrieved from multiple sources need to be combined to get one answer. As a summary, complex questions are complex because:

- The questioner has implicit knowledge that is not part of the question and therefore is not known to the QA system; The context has to be known by the QA system. In a dialogue system a clarification question could be posed to the user when it is recognized that more information is needed. In the context of a 'pure QA system' this can not be done and another approach is needed.
  - For example: The questioner can have a presupposition, that is either correct or not.
  - If the presupposition is explicitly stated, the question becomes a composed question.
- The question is composed of a number of other questions. It has to be recognized by the system that this is the case and the question has to be split up in some way so that a number of queries can be sent into the rest of the system.
  - The question is difficult to analyse syntactically or semantically.
  - There is a need for judgement to find out which questions the original question should be decomposed in.
  - The scope of the question is important. How broadly or narrowly should the question be interpreted.
- The answer has to be constructed from a number of documents (or even multi-media or multi-lingual sources). Interpretation of the received information is needed. This seems to be mainly a problem for the answer-side of the QA system, although queries probably will have to be adapted for different media or different languages.
  - Composed questions are in any case returning a number of facts that have to be combined into one answer, but for example answers to list questions or definition questions that can be answered by using a template, also have to be constructed from multiple documents.

The number of complex questions could be endless, but some are (partly based on Manning, 2004 and Saquete et al., 2004b):

- Definition questions (only partially complex when templates can be used)
- List questions (only complex on the answer side)
- How-many questions (more or less the same as list questions in that a list could be retrieved and the number of items counted)
- Script questions, like *'How do I assemble a bicycle?'*
- Questions with temporal relations
- Questions with causal relations (presuppositions, could be not explicitly stated)
- Questions with logical connectives
- Questions with evidential relations
- Questions with part-whole relations

3.3 Choice of complex questions to handle

In 2003 and 2004, the DFKI has participated in the QA track of CLEF. As mentioned in Section 2.2, the task for 2005 comprises the handling of 'temporally restricted questions'. This made it very appealing to choose these questions as the type of complex questions to deal with in this thesis.

With complex questions, regardless which kind of complex questions, always seems to come the need for splitting the questions into multiple questions and fusing the answers together into one answer. This is why the question side of a 'normal, all-purpose' QA system (question analysis, splitting of questions) can not be changed solely, without changing the answer side (answer extraction, answer fusion). Because this is needed for each type of complex questions, it seems that when a system has been adapted for one type of complex questions, the step of upgrading the system to handle a next kind of complex questions, should be 'relatively easy'.

For example, questions containing logical connectives, like *'Which country borders Germany and Belgium?'* can be split into two questions *'Which countries border Germany?'* and *'Which countries border Belgium?'*. After that the countries (or country) that are in both sets of answers are the answer for the original question, just like a question like *'Who was Russia's president in 1999?'* can be split in *'Who was Russia's president?'* and *'in 1999'* and combining the answers of the first question (together with the respective periods in which they were president) and the given date, leads to the answer. This is further elaborated on in Section 5.1.

Because there was a good reason to choose 'temporally restricted questions' to deal with (because of CLEF) and it was assumed and hoped that different types of complex questions can be dealt with in the same way, it was decided to choose those 'temporal questions' to handle in this thesis.

4 Relevant state-of-the-art in answering complex questions

In the second chapter, it was explained how QA systems have moved from answering just simple factoid questions to answering complex questions. In the previous chapter, it was explained what complex questions are and a choice was made for one type of complex questions to handle in the rest of this thesis, namely temporal questions.

In this chapter the relevant state-of-the-art in answering complex questions is discussed. The only group that has treated temporal questions before is the group of Saquete et al. (2004a, 2004b). Their work is discussed in the first section. The next three sections describe the approach of three other groups that have dealt with complex questions and are relevant for this thesis. In the last section, Section 4.5, it is concluded what can be learned from these groups.

4.1 Saquete et al.

Saquete et al. (2004a, 2004b)[17] use a multi-layered system for answering complex temporal questions in CLEF 2004. The questions are split up in multiple questions that are sent into a 'general purpose QA system'. The main advantages of this multi-layered system are, according to Saquete et al.:

- It allows to use any existing general QA system, with the only effort of adapting the output of the processing layer to the type of input that the QA system uses.
- The QA system does not have to be modified when in the future additional types of complex questions are being handled. Only a new layer has to be constructed.
- Each additional layer is independent from the others and only processes the type of questions accepted by that layer.

Four kinds of temporal questions are distinguished, of which two types are simple and two complex. The distinction is made, based upon the presence of:

- Temporal expressions: A moment in time, for example a year, a period or an 'event' like *'the Second World War'*.

[17] They were already mentioned in the last chapter.

- A temporal signal: The signal establishes the order between the events in the question (in this case an event can also be a date or period). For example *after* in *'Who was president after 1900?'*.

The four types are:

1. Single event temporal questions without temporal expression (simple, no temporal signal, no temporal expression):
   'When did Iraq invade Kuwait?'

2. Single event temporal questions with temporal expression (simple, no temporal signal, temporal expression (1988)):
   'Who won the 1988 New Hampshire republican primary?'

3. Multiple events temporal questions with temporal expression (complex, temporal signal (after), temporal expression (in August 90)):
   'What did George Bush do after the U.N. Security Council ordered a global embargo on trade with Iraq in August 90?'

4. Multiple events temporal questions without temporal expression (complex, temporal signal (before), no temporal expression):
   'Where did Bill Clinton study before going to Oxford University?'

A decision tree is used to decide of which type a question is. How this is implemented is not explained in Saquete et al. (2004a, 2004b). The complex questions are then split up in two questions. The part before the temporal signal is a sentence on its own, but the part after it is not. Therefore the second part is changed slightly to form a correct sentence. For the last two questions mentioned above the questions become:

3 *'What did George Bush do?'* and *'When did the U.N. Security Council order a global embargo on trade with Iraq in August 90?'*.

4 *'Where did Bill Clinton study?'* and *'When did Bill Clinton go to Oxford University?'*.

In both cases, the first question is a list question and the second a simple, factoid question.

It is important to notice that in CLEF 2005, the temporal questions are classified differently (see Section 2.2). The first type distinguished by Saquete et al. is classified as a normal factoid question (with expected answer type DATE or PERIOD). The second type is type 1 and 2 of the QA@CLEF, questions restricted by date or period. The fourth type of questions distinguished by Saquete et al. are the type three questions distinguished in CLEF. The type three is somewhat more difficult. Type 1, 2 and 3 questions in CLEF

can contain a temporal expression as well as a temporal signal and would therefore be classified in the third category by Saquete et al.. For example *'Who was president after 1994?'* and *'Who was president after the war in Iraq in 1994?'*.

The results they had on the splitting of 112 of the 123 temporal questions from the TERQAS question corpus[18] are shown in Saquete et al. (2004b) in Table 2 and is here shown as Table 2. They discarded 11 questions, because they needed a treatment beyond the capabilities of their system (further information in Saquete et al., 2004b). Notice that these results only cover the splitting of the question, not the extraction of the answers or the composition of the answer from multiple answers[19].

In Table 2: 'TE Recognition and Resolution' is about the recognition of temporal expressions, of which there where 62 in the corpus. Type identification is about the recognition of the correct types (their types, not the CLEF types, as explained above). Signal detection means if the system finds the 'temporal signal' in a type 3 or 4 question (the two complex ones). The number of complex questions correctly split is in 'Question Splitter'. The overall performance of the decomposition unit can be seen in the last row.

	Total	Treated	Successes	Precision	Recall	F-Measure
TE Recognition and Resolution	62	52	47	90%	75%	86%
Type Identification	112	112	104	92%	100%	93%
Signal Detection	17	14	14	100%	82%	95%
Question Splitter	17	14	12	85%	71%	81%
Decomposition Unit	112	112	93	83%	83%	83%

Table 2 Evaluation of Saquete et al.'s system

4.2 Prager et al.

As mentioned in Section 2.3, Prager et al. (2003, 2004) use QA-by-Dossier-with-Constraints (QDC) to find out that the most likely answer their system returns to the question *'When did Leonardo da Vinci paint the Mona Lisa?'* can not be *2000*. Their system knows that when someone delivers a work, the date of delivery must be after his date of birth and before his date of

[18]Two links to the corpus are given by Saquete et al., one of which still works (June 6, 2005), being http://www.cs.brandeis.edu/~jamesp/arda/time/documentation/TimeML-use-in-qa-v1.0.pdf. The other link given is http://time2002.org

[19]Because the corpus was not available, it could not be used to compare the results of Saquete et al. with those the system under development would have.

death. Therefore, the right answer turns out to be *1503*, which is the fourth most likely answer.

As a start, this system used QA-by-Dossier (QbD, without the constraints), which means the system asked auxiliary questions, assembled the answers to these questions into a dossier and returned that to the user. In the QDC, the answers to the questions are checked for consistency with each other. This QbD and QDC can be used perfectly for answering definition question, but as shown by the 'Da Vinci example' above, it is also very useful for other questions.

Another interesting part of their system is their use for different agents for different questions. Although generic methods are of course appealing, it seems this is not possible in question answering.

4.3 Moldovan et al.

Moldovan et al. (2004) describe two QA systems with a lot of interesting (new) features, but for question analysis of complex questions, there is really just one interesting feature: the use of an architecture that uses a number of different strategies, depending on the recognized question type. After the question type is recognized, one strategy (or a couple of strategies) is (are) selected and used to find an answer. It is also described how they handled for example anaphora and ellipsis resolution, but this in the context of TREC, where first a topic is presented and then a couple of questions on this topic are posed. In this case, the ellipsis can be resolved using the topic. This is not the case in the CLEF context and can not be used to solve the ellipsis in for example *'Who was queen before Juliana?'* (which means *'Who was queen before Juliana was queen?'*).

4.4 Diekema et al.

Diekema et al. (2003) describe a system that is used by students to ask questions in the domain of Reusable Launch Vehicles (aerospace engineering)[20]. The questions posed are more complex than for example TREC questions and can often not be directly answered. The system they propose can ask the user for clarifications, just like a librarian would do. The clarification question can be a simple yes/no question or a more complex question. Although this idea seems promising, it can not be used in question answering as done in the CLEF and TREC forums (because systems are not allowed to interact with questioners) and therefore, their approach is unfortunately not very useful in the context of this thesis.

[20]See also the last chapter, where their notion of complex questions is discussed.

4.5 Conclusions

An important and overall conclusion is that there are not too many groups dealing with the types of complex questions relevant for this thesis. There is only one group that has dealt with the types of temporal questions dealt with in this thesis.

The idea, by Saquete et al. (2004a, 2004b), of constructing a layer surrounding a 'normal all-purpose QA system' that deals with temporal questions seems very interesting. It is easier to test what kind of QA system functions best when trying to answer temporal questions and it is generic in that other kinds of questions can be easily added as different layers.

Using a decision tree to distinguish between the four types of temporal questions (these four kinds seem logically defined) and using different strategies on each kind is necessary. Using different strategies is just like Moldovan et al. (2004) and Prager et al. (2003, 2004) do.

The idea behind the QA-by-Dossier-with-Constraints, used by Prager et al. (2003, 2004), could be interesting for temporal questions in that answers could be checked for being possible.

5 Design

In this chapter, the design is explained of the system developed in the context of this thesis. In the first section, the general approach is outlined. In Section 5.2 it is explained in which syntactic ways the temporal restrictions in temporal questions can come and how the different ways are handled in the system. The chapter is finished by Section 5.3, where the internal representation of extracted dates and periods is elaborated on.

5.1 General approach

The general working of the system is described in this section. It starts with an overview of the mono-lingual system and then it is shortly explained what needs to be changed in order for the multi-lingual system to work. This is all in the first subsection. In Subsection 5.1.2 it is explained which two different strategies are used and why. This section is ended by Subsection 5.1.3, where an example is given of a question that is dealt with by the temporal QA system.

For the understanding of this section it is important to recall that in CLEF there are three types of temporal questions, as seen in Section 2.2. These types are used here. The questions are temporally restricted by:

1. Date
2. Period
3. Event

The part that restricts the question temporally is called the restricting part, the remaining part of the question is called the 'real question' or the stripped question, because it is the main question, stripped from its restriction.

5.1.1 Overview

As seen in the last chapter, it is probably a good idea to have different strategies for different kinds of questions. The way Saquete et al. (2004a, 2004b) use a kind of 'layer' constructed round a 'normal QA system' is interesting. Therefore it was decided to construct the 'temporal part of the QA system' in the same way (from now on this part will mostly be just called 'temporal QA system', the 'normal, all-purpose QA system' will be called 'normal QA system' or just 'QA system'). Questions the QA system receives are to be checked by the temporal QA system to see if they are temporal questions and if so, the temporal QA system starts processing them. If not, they are sent to the 'normal QA system'. This makes it very easy to 'plug in' other layers. In the context of CLEF the check is not necessary, because the temporal questions are tagged as to their type.

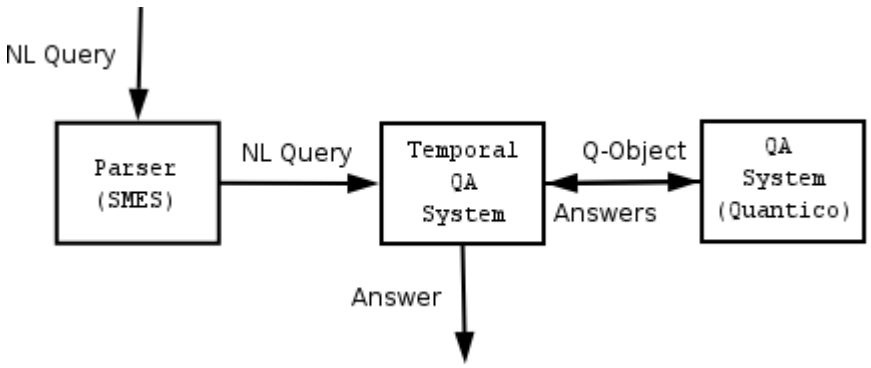

Figure 1 The 'temporal' QA system using the 'normal' QA system

The general flow of data that results from a temporally restricted question is shown in Figure 1. A question in natural language is parsed by SMES[21], or theoretically any other parser, and a Question-Object (Q-Object) results. This is an object containing the keywords extracted from the question and some other information, like for example the expected answer type (EAT). A question like *'Who was President in 1988?'* would (hopefully) result in the keywords PRESIDENT and 1988 and the EAT PERSON. The temporal QA system changes the Q-Object in the way explained below and uses the 'normal, all-purpose' QA system Quantico, or theoretically any other 'all-purpose' QA system, to retrieve answers. The temporal QA system decides which answer should be returned to the user, in the way described below, and the answer is returned.

For the multi-lingual context, the original question is first parsed by SMES, and then translated. Only then the processing by the temporal QA system is started. This means the temporal QA system does not know it is working multi-lingual and exactly the same approach can be used[22]. It could be interesting to receive the original question by the temporal QA system, split it (when necessary, as shown in the next subsection) and then translate both parts on their own. The translation could be better in this case, but the problem is, the second part of a split question is not a correct question anymore and it has to be generated. When the original question is *'Who was queen during the Second World War?'* the question is split into *'Who was queen?'* and *'The Second World War?'*. This second part obviously needs some modification before it becomes a real sentence. Therefore, this approach of translating after splitting, was not chosen. In Section 7.5, where

[21] See Neumann and Piskorski (2002) for the working of SMES.

[22] In the next chapter, where the implementation is elaborated on, it is shown there are some small differences for the temporal QA system, for example due to non-perfect translations.

the system is evaluated, it can be seen that the translation of a complete, non-split question does not cause more problems than the translation of both parts independently.

5.1.2 Two different strategies

As with each complex question, as seen in Chapter 3, temporal questions (temporally restricted questions) can be split into a couple of questions and processed step by step, by first retrieving the different answers and then combining them into one answer. All different six characteristics mentioned by Carbonell et al. (2000), as seen in Section 3.1, come into play here, when it should be decided how to split the questions and how to construct one answer out of the multiple facts. However, a lot of questions could probably be answered without this decomposition. Therefore, two different strategies are chosen (for both the type 1 and 2 questions and the type 3 questions):

1. Direct method: The Question-Object is directly sent into the 'normal QA system', most importantly the keywords contained in the question, together with the expected answer type (EAT).

   A problem is: Which keywords should be sent. When for example the question is *'Who was queen before Juliana?'* searching for QUEEN, JULIANA and EAT PERSON probably does not lead to an answer: such a number of sentences is retrieved that precision would be very low. The addition of *before* could lead to a passage containing the right answer. Unfortunately this leads to some other problems:

   - Searching for a name in the passages, the sentence *'Juliana was queen before Beatrix'* would lead to the answer *Beatrix*, which is obviously not the correct answer.

     This problem can be solved using the 'syntax-trees' of both the question and the answer, including the terms searched for. In that case the aforementioned question has the same tree as *'Wilhelmina was queen before Juliana'* (notice that *Who* is not a query term, because it is not a 'keyword', of course it is an important word to decide what the EAT is), but *'Juliana was queen before Beatrix'* does not have the same tree, because the query term *Juliana* is in the wrong place.
   - When searching for an answer with the terms QUEEN, JULIANA and BEFORE, it would not be noticed the sentence *'After Wilhelmina, Juliana became queen'* also contains the correct answer.

     This problem is somewhat more difficult to cope with. The first part of the solution would be to add some keywords. Instead of a query like QUEEN AND JULIANA AND BEFORE, a dictionary

would be used and create a query like QUEEN AND JULIANA AND (BEFORE OR AFTER OR ...).
However, another problem arises. The syntax-trees are not the same anymore, because of, in this case, the change of *before* to *after*. The expected tree in the answer sentence could be changed when *after* was found instead of *before*, however, this seems difficult. An easier solution seems to represent the knowledge asked for internally and check each answer if it has the same representation. The question would become PERSON? < JULIANA. The answer *'After Wilhelmina, Juliana became queen'* can be represented as JULIANA > WILHELMINA, so it can easily be deduced this is the correct answer. More on this can be found in Section 5.3.

The example used was that of a type 3 question, however, exactly the same approach can be used for a type 1 or 2 question.

It was decided that this strategy did not come in the first place, so this strategy is not really worked out. This means for type 3 questions that for the example above the keywords QUEEN, JULIANA and BEFORE are used and an answer is only found when the answer sentence contains the word *before*. Also, it is not checked if the 'syntax trees' are correct, so in the example given above, *Beatrix* could be seen as the correct answer. For type 1 or 2 questions, this does not cause real problems. The temporal part is extracted, even when this direct method is used, and the dates in possible answer sentences are compared to this date. However, the problem that *before* should be in the answer sentence is still there. Leaving this out, actually leads more or less to the next strategy, so having BEFORE as keyword will not lead to a lower than necessary recall because the next strategy will find the answer. This first strategy will probably lead to a high precision, where the next strategy leads to a higher recall.

2. Question decomposition: If the direct method does not lead to an answer, a question like *'Who was queen before Juliana?'* could also be split into two questions like Saquete et al. (2004a, 2004b) have done. The question would ideally be decomposed into *'Who was queen?'* and *'When was Juliana queen?'*. The answers to the first question however are not useful if it is not known when each person was queen. So the question becomes really *'Who was queen and when was this person queen?'* The answer to the first question will be a list with queens and the time spans in which they reigned. By using the answer to the second question, the correct answer can be found. Although this method seems very simple, it is actually not that simple:

   - How can the system know Juliana was a queen?

Ellipsis resolution is needed. In this question, it can be argued that the question actually is *'Who was queen before Juliana was queen?'*, so the original question has to be expanded. It is even possible to check the correctness of the ellipsis resolution by checking Juliana was a queen. This could be done by asking a yes/no-question *'Was Juliana queen?'* or by asking for a definition of *Juliana* and checking if it is mentioned she was a queen.

A problem closely related can be seen in the example *'Where did Bill Clinton study before he went to Oxford University?'*. When this question is split, the temporal part is *'before he went to Oxford University'* and it is not clear who *he* is. Anaphora resolution is needed here.

In the system, the ellipsis resolution is not taken care of (the anaphora resolution is). In a lot of cases this will not cause too much trouble, since for example the example given above (with the queens), can hopefully be answered by the direct method. Only when the temporal restricting part has not too much to do with the main clause, the splitting could be really necessary. In such cases it is not possible for a questioner to use ellipsis, because this can only be used when in both sentence parts information is the same and can therefore be omitted.

- How can the system know the questioner is looking for queens in the Netherlands and not, for example, in the United Kingdom?

  This is some context knowledge the system has to retrieve, a characteristic of complex questions, as seen in Chapter 3. This knowledge could be retrieved from the sentence, because the other queen mentioned is also Dutch. The problem is that the system does not know this. The system could ask for a definition of *Juliana*, but which part of the retrieved definition is interesting? Why is the country she comes from important? People know this, because of their 'internal ontology', but for open-domain, this can not be known.

  Another possibility could be to check the country the question has come from (when using such a system on the Internet). This solution however is definitely not full-proof. At the moment of writing there is no better solution than to hope for a 'better' question posed or the direct method finding the answer (both of which are realistic options).

- Another problem is that, when for example the question (originally in Saquete et al., 2004b) *'What did George Bush do after the U.N. Security Council ordered a global embargo on trade with Iraq in August 90?'* is split, the questions become *'What did George Bush do?'* and *'When did the U.N. Security Council or-*

der a global embargo on trade with Iraq in August 90?'. The first question does not seem to be a very useful question, George Bush probably did thousand of things in his life.

A solution for this problem has not been found yet.

The example used was that of a type 3 question, however, exactly the same approach can be used for a type 1 or 2 question. The mentioned problems will not be problems there, because there will be no ellipsis for example.

When a date is received for a type 3 question, the result can be treated as a type 1 question (or a type 2 question if a period was retrieved) and an answer can again be first tried to be found by the direct method and secondly by decomposing the question. The example in the coming subsection will clarify this.

5.1.3 Example

When the question is *'Who was president of the United States when the Berlin Wall was torn down?'*, it is of type 3. The question is received by the temporal QA system and is directly sent into the normal QA system. Answer sentences are retrieved, but an answer is probably not found and therefore the question is decomposed.

The second part of the sentence is the restricting part and is transformed into a question with expected answer type DATE. The resulting question, *'When was the Berlin Wall torn down?'* is sent to the 'normal QA system'. An answer is received from the system and is hopefully *1989*. Now, the total question can be rewritten as *'Who was the president of the United States in 1989?'*, a type 1 question. This question is sent back as if it were a question directly received from the user. Maybe an answer is now in the sentences received from the QA system, maybe not.

In the first case, the answer is given to the questioner. In the second case, the type 1 question is split again and the question *'Who was the president of the United States?'* is sent. Together with this sentence (or actually the keywords from this sentence), the EAT PERSON and a restriction telling the QA system to only send back sentences containing a DATE, are sent. This is done because the question the temporal QA system really wants to send into the 'normal QA system' is *'Who was the president of the United States and when was he president?'*. A possible answer sentence that does not contain a date can not be checked against the temporal restriction in the original sentence or the date received for the restricting part of the original sentence. In this example, the dates in the retrieved possible answers are

checked against *1989* and the correct answer is extracted and returned to the questioner.

5.2 Possible temporal restrictions

Temporal restrictions are not all the same in syntax and grammatical role. The different ways a temporal restriction can be found in are described here, partly based on Pustejovsky et al. (2005).

For a type 3 question, the temporal restriction can come as:

- A subordinate clause, connected by a subordinating connective to a main clause as in for example: *'Who was queen before Juliana (was queen)?'* where *'Who was queen'* is the main clause and *before* is the connective connecting *'Juliana (was queen)'* to the main clause.
- A prepositional phrase in for example *'Who was president during the last World War'*, where *'during the last World War'* is the prepositional phrase (PP). The noun has a meaning that has to do with an 'event'. Pustejovsky et al. (2005) call these 'event nominals'. The complete PP is the restricting part. A noun that is not in a PP can not be the restricting part in German and English (for English see for example Pustejovsky et al., 2005).

For a type 1 or 2 question, the temporal restriction can not be a subordinate clause, but there are two possibilities:

- It can be a prepositional phrase acting as an adverb. The head of the prepositional phrase should be a noun (phrase), representing a date. For example: *'Who was president in 1940'*, where *'in 1940'* is the prepositional phrase, acting as an adverbial phrase.
- An adverbial phrase or adjective could directly be a date. For example *'Who won the 1988 European championship?'*, where *1988* acts as an adjective. Another example is (in German) *'Wer war 1940 Präsident?'*, in which *1940* acts as an adverbial phrase.

It is important to notice that not all constructions mentioned here are correct for both English and German and that they are surely not correct for all languages of the world. Other languages probably have other possible constructions. In this thesis the focus is on English and German, so other languages are not looked at.

So there are really three categories of temporal expressions. This is summarized in Table 3.

Type of restriction	Type 1: Date	Type 2: Period	Type 3: Event
Subordinate clause			*
Prepositional phrase	*	*	*
direct date	*	*	

Table 3 Types of possible temporal restrictions for each question type

For each type of restriction it is shown what parts of it are interesting and why:

1. Subordinate clause: The subordinating connective gives the type of relation, for example BEFORE or AFTER, and the subordinate clause indirectly represents a date.

2. Prepositional phrase: The preposition gives the type of relation and the noun (with 'event' meaning) indirectly represents a date or period or a date or period is directly there.

3. A date can be used on its own. The relation is then something like EQUAL or INCLUDES. The date is represented trivially.

A date on its own will be tagged by the Named-Entity tagger, that tags each question before it is parsed by SMES, and are therefore easy to extract. However, the first two categories mentioned above cause problems in deciding what the temporal relation is.

1. Questions are parsed by SMES and subordinate clauses can be extracted. A problem is that subordinating connectives can have multiple meanings. For example: *wenn* in German, can be used to express a non temporal condition in *'Was hätte Clinton gemacht, wenn er noch Präsident gewesen wäre?'* (*'What had Clinton done, if he were still president?'*), but it can also have a temporal meaning in *'Was wird Bush machen, wenn der Krieg vorbei ist?'* (*'What will Bush do, when the war is over?'*). In the implementation a practical solution is used for this problem, see Section 6.3, but a real solution is not implemented yet.

   In German it could be used that the verb is in the 'Konjunktiv Irrealis', but that only functions for this example. However, the solution should probably be sought in this direction. Conjunctives are closed class words. For English, it is relatively clear when they have temporal meaning[23] but when it is not clear like in the example above, per word

[23]See for example http://www2.gsu.edu/~wwwesl/egw/bryson.htm, last checked on July 31, 2005.

the meaning should be clarified. So for the example above, it could be checked if the verb is in the 'Konjunktiv Irrealis'. Another word like *als* in *'Welche Stadt is großer als Kohl gedacht hat?'* (*'Which city is larger than Kohl had thought?'*) or *'Wer war König als der Krieg endete?'* (*'Who was king when the war ended?'*) needs a different check. For the first use of *als*, a comparative has to be in the sentence, so this could be checked. For each conjunctive with multiple meanings a couple of different checks should be implemented.

2. The same problem cause prepositions. For example: *in* can have a temporal meaning in *'Who was president in 1990?'* and a non-temporal in *'Who was president in Germany?'*. When a preposition has multiple meanings, it is checked if the head of the prepositional phrase is tagged as date or it is a noun with an 'event' meaning, like *war*.

   Finding prepositional phrases is done by using the parser that parses the question and finds the PP. Checking if a noun has event meaning is done by using a lexicon containing nouns with event meaning.

Notice that with the proposed differentiation between the different types the correct restricting part will (theoretically) be extracted for some more difficult examples:

- In a question like *'Who received the 1989 Nobel Prize for Peace?'*, there is no PP or subordinate clause, so it is a type 1 question. Although *1989* is a qualifier of *'Nobel Prize for Peace'*, only *1989* should be extracted.
- When the question is *'Who received the prize during the 1989 Nobel Prize elections?'*, the complete PP *'during the 1989 Nobel Prize elections'* should be extracted and it would be a type 3 question.
- When a noun with 'event meaning' is found, but it is not in a PP, it will not be extracted. In for example *'When was the election?'*, it can be seen that this is correct, since *election* should not be extracted.

The temporal QA system does not cope with, as Pustejovsky et al. (2005) call them, intensionally specified expressions like *three years ago* or *last month*.

5.3 Internal representation of dates

When a question is split, be it a type 1, 2 or 3 question, the result is a 'temporally unrestricted question' and a date or period. When the question was a type 1 or 2, this date is trivially received. When it was a type 3

question, it is the date or period that represents the event in the question. This date was received by asking the QA system when the event took place.

The date or period (temporal expression) needs to be stored and the found answer sentences are checked against this expression. For this comparison of temporal expressions, not only the date or period itself is important, but also the preposition or subordinating conjunctive that expresses the type of relation. For example: The sentence *'Who was queen before 1940?'* could give a total other result than *'Who was queen after 1940?'*.

Combination of dates is done by using categories of relations, for example BEFORE, AFTER and EQUAL. This is based upon work by Pustejovsky et al. (2005) and Schilder and Habel (2001), who base this system on among others Allen (1983a, 1984) and Mani and Wilson (2000). Pustejovsky et al. and Schilder and Habel use respectively thirteen and seven types of relations. However, for this thesis it was decided to use only the three types of relations mentioned above. Other types distinguished by Pustejovsky et al. are for example 'one being the ending of the other' (*'John stayed in Boston till 1999.'*) and 'one including the other' (*'John arrived in Boston last Thursday.'*).

The difference between these papers and this thesis is that they were not written in the context of QA, but are focusing on semantic tagging. In the context of QA, the three mentioned relations seem to be sufficient, because it is already difficult enough to decide between these three types when deciding to which relation a temporal expression belongs.

Internally the dates found in the questions and (possible) answer sentences are represented with $<$ for BEFORE, $>$ for AFTER and $=$ for EQUAL. The dates are represented loosely like ISO-8601 1997 as proposed by Mani and Wilson (2000). This has led to the representation YYYY:MM:DD. When month and/or day are not known, the representation becomes YYYY:MM or YYYY. For example, the temporal expression *'after May 3, 1945'* is internally represented as $>$ 1945:05:03. Comparison of dates is very straightforward: it can easily be seen that the answer with temporal expression $=$ 1945:02 is not the correct answer for the question containing $>$ 1945:05:03.

For periods, nothing really changes, except that for example *'between June 12, 1955 and July 1965'* is represented as $=$ 1955:06:12 – 1965:07. Theoretically also something like $>$ 1945 – 1946 can be saved, although this does not seem very useful. *'in the summer of 69'* could be saved as $=$ 1969:06 – 1969:08, but instead it is just saved as $=$ 1969. Summer is not really exact: maybe the user even comes from the Southern hemisphere.

6 Implementation

In the last chapter the design of the temporal QA system, the part of the total QA system that acts as a layer surrounding a 'normal QA system', was explained. In this chapter it is shown how this design is implemented.

In the first section it is shown how the system is connected to the existing QA system and which consequences this has had. In Section 6.2 the requirements posed by the QA system are elaborated on, together with their consequences on the implementation of the temporal QA system. In the last section, Section 6.3, some other choices are explained, where the implementation does not completely follow the design.

6.1 Connecting to the existing QA system

The diagram in Figure 2 shows how the temporal part part of the QA system is connected to the rest of the system[24]. In this section it is explained how the complete system functions together. In Figure 3 this is expressed in pseudo-code to clarify the working. The temporal part is called the temporal QA system.

An incoming question gets named-entity annotated by LingPipe[25], and parsed by SMES[26], a syntactic parser for German, constructed at the DFKI. SMES constructs a Q-Object (question object) that among other things, consists of the keywords from the question, the original question sentence and the expected answer type.

Normally now the sentence would have been sent to the temporal QA system, but in CLEF, the question is tagged, so only temporal questions are sent to the temporal QA system. The temporal QA system lets the 'Lucene query interface', that is a part of the 'normal QA system', construct a query for the Lucene document retrieval server[27]. The temporal QA system sends the query to Lucene, receives the possible answer sentences and decides if the answer is already found. How this is done exactly for type 1 or 2 questions can be seen in Section 5.1. For type 3 questions a possible approach can be seen in the same section, but it is implemented somewhat 'easier': if the number of extracted answer sentences is greater than a set threshold, the

[24] How the rest of the QA system functions, is explained in more detail in Neumann and Sacaleanu (2005).

[25] Documentation and free download on http://www.alias-i.com/lingpipe/, last checked on July 10, 2005.

[26] Documentation and free download on http://www.dfki.de/~neumann/pd-smes/pd-smes.html, last checked on July 10, 2005. See also Neumann and Piskorski (2002) on the working of SMES.

[27] Documentation and free download on http://lucene.apache.org/, last checked on July 10, 2005.

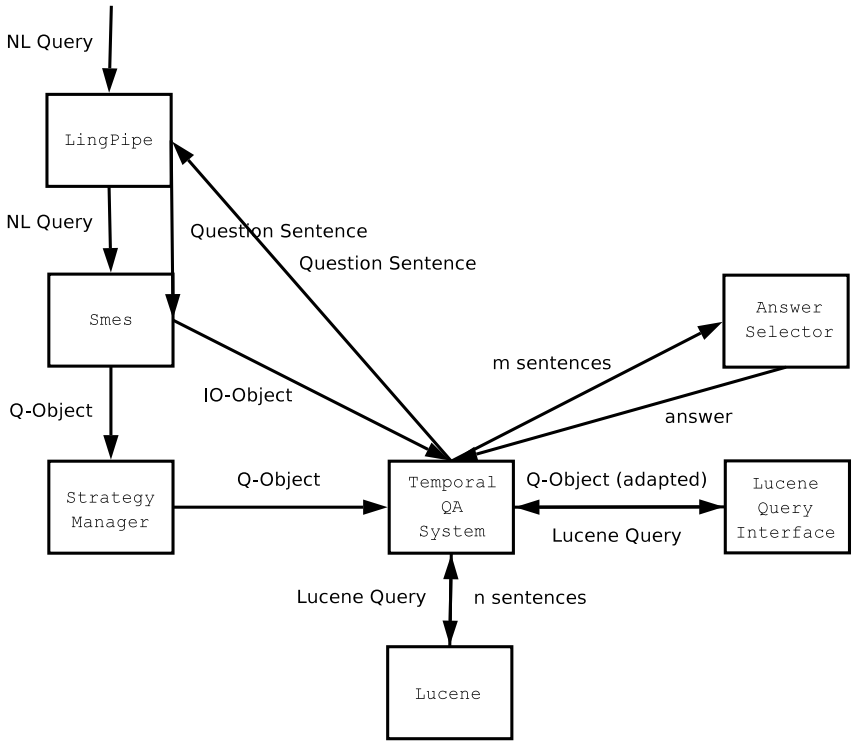

Figure 2 The temporal part connected to the 'normal' QA system

most probable answer is retrieved from the answer selector and returned.

If the answer is not already found, it depends on the type of question what is done next. With a type 3 question, a new Q-Object, containing just the temporal element that was in the original question is sent to the Lucene query interface. Because in order to extract this temporal restricting part, only the keywords do not suffice, LingPipe and SMES are used to get an IO-Object (information object containing a syntactic parse-tree) of the question so that the relevant prepositional phrase or subordinate clause can be extracted. The received query is sent and the possible answer sentences are received. These are directly sent to the answer selector and the most probable date is received, just as if the system had received a normal question with EAT DATE. This date is added to the Q-Object that was stripped from its temporal restriction and handled like a normal, directly received type 1 question (see above). When no answer is found to the 'date question', the systems halts here. It now depends upon whether or not an answer was found to the originally sent query (directly from the type 3 question) if any answer will be returned.

If the answer of a type 1 or 2 question is not directly received, a Q-Object

is constructed not containing the temporal part. Again, a Lucene query is constructed and the temporal QA system adds a term to the query so that only documents containing a date are extracted. The possible answer sentences are internally compared to the extracted temporal part (as explained in the last chapter) and the sentences that are found to be correct are sent in order to retrieve the most probable answer from these sentences. This answer is returned to the questioner.

```
procesQuestion(qObject) {
  LuceneQueryInterface.getQuery(qObject)
  Lucene.getPossibleAnswers(luceneQuery)

  if(type == 3) {
    if(answers.size()>treshold) {
      return QAsystem.getBestAnswer(answers)
    } else {
      split(qObject)
      LuceneQueryInterface.getQuery(temporalQObject)
      Lucene.getPossibleAnswers(luceneQuery)
      QAsystem.getBestAnswer(answers)
      constructQObject(strippedQObject,date)
      return procesQuestion(type1or2QObject)
    }
  }

  else if(type == 1 || type == 2) {
    checkTemporalRestriction(possibleAnswers)
    if(checkedAnswers.size()>treshold) {
      return QAsystem.getBestAnswer(checkedAnswers)
    } else {
      split(qObject)
      LuceneQueryInterface.getQuery(strippedQObject)
      luceneQuery.add(neTypes:DATE)
      Lucene.getPossibleAnswers(luceneQuery)
      checkTemporalRestriction(possibleAnswers)
      return QAsystem.getBestAnswer(checkedAnswers)
    }
  }
}
```

Figure 3 Pseudo-code of the implementation of the complete system

6.2 Requirements posed by the existing QA system

A number of restrictions to the system are posed by the existing QA system that influence (the performance of) the temporal QA system.

The output from the syntactic parser is in a special format, not supported by parsers freely available (other than SMES of course). Although it is possible to build some sort of converter between for example Lexparser[28], that can parse German and English, and the wanted format, this would take a lot of time. Therefore, the system is now build such that English questions are first translated and then parsed by SMES, so that when German answers are searched for, the system effectively works mono-lingual. German sentences are directly parsed and only afterwards translated.

The IO-Objects that SMES can construct are not very deep. This means that when a sentence is received like *'Who was president after the invasion of Iraq in Kuwait?'* the system can not know where the subordinate clause ends. The start is easy because of the subordinating conjunctive. In this case, it is no problem, because the subordinate clause ends with the sentence ending, but a sentence could also look like *'Who was, after the invasion of Iraq in Kuwait, president?'*. In this case, the system can not determine which part is the temporal part and extracts the complete part starting with *after*. Although such a sentence could seem a little unusual, in other languages, like Dutch, this is very common.

When relevant pieces of text are searched for with keywords and an expected answer type, complete documents could be returned when they satisfy the query, but this is generally not a good approach in QA. Most systems work on some sort of 'passages', that are a couple of sentences long, or just on sentences. The system at the DFKI uses only sentences, which influences recall. However, precision could be higher as a result of this.

When normally a factoid question starting with *When* or in German *Wann* is posed, it can not be deduced from only this word if the EAT is DATE or PERIOD. This can only be deduced from the meaning of the verbs or nouns in the sentence. For example: *'When was Clinton president?'* or *'When did Clinton marry?'* clearly have a different answer type for humans, because of their 'internal ontology'. The QA system used does not have such an ontology and always uses EAT DATE. When the temporal QA system is trying to construct a type 1 or 2 question from a type 3 question, it asks for a date or period for the restricting temporal part. Because of the functionality offered by the normal QA system, this will always be a date, so the resulting question is always a type 1 question.

[28]Documentation and free download on http://www-nlp.stanford.edu/software/lex-parser.shtml, last checked on July 10, 2005.

6.3 Other implementation choices

In the last two sections it was showed which consequences it had for the temporal QA system that it was connected to an already existing QA system at the DFKI: Quantico. Internally in the system, some other choices have been made. The class diagram that results from this is shown in Figure 4.

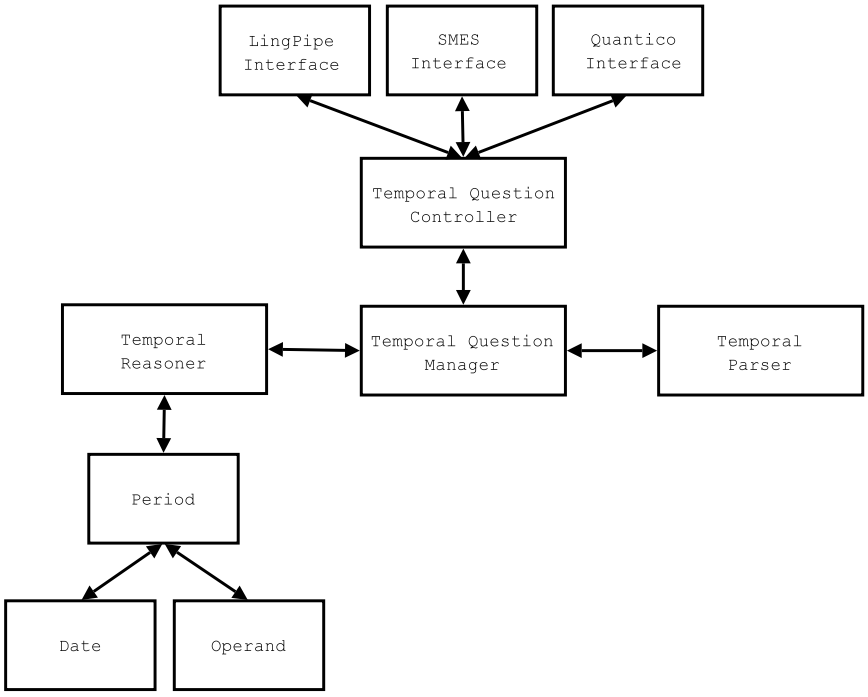

Figure 4 The temporal part of the QA system

The temporal question controller only controls where information should flow to and uses three interfaces to connect to three different servers. The quantico interface handles all the information flows concerning the retrieval of answers from possible answer sentences, the construction of queries, the initial sending of a question and the eventual reception of an answer. The temporal question manager decides what is to be done with a question, constructs Q-Objects and decides which answer is returned. It uses the temporal parser to find out what part of the sentence the temporal part is and uses the temporal reasoner to compare the extracted temporal expression with dates found in the answer sentences.

Because in CLEF it is already known the questions are temporally restricted, the checks mentioned in the Section 5.2 are somewhat loosened. This means that if no prepositional phrases are found and just one subordinating connective (and no dates), this connective is said to be the beginning of the

temporal restricting part, even if the system can not deduce this from anything it knows. The same holds if no connectives are found and just one prepositional phrase with multiple possible meanings. When the system can not determine a temporal restricting part and there are multiple possibilities, for example two prepositions that could both have temporal meaning, the first possibility is used. The idea behind this is that this gives a 50% change, where just giving up gives a 0% change of using the right prepositional phrase.

7 Evaluation

Because of the rash implementation to get the system ready for CLEF, it turned out there were some bugs and some problems not thought of. In the first section these are summed up and solutions are shown. The results the system had in CLEF and the results the system would have had if those mistakes had not been in the system in the first place are in the next section, Section 7.2.

In Section 7.3 it is analysed what the performance is of this way of handling the complex temporal questions: is it useful when answering complex temporal questions to deal with them as proposed and implemented in this thesis?

In Section 7.4, it is analysed how the system performs in splitting questions, so extracting the temporal restriction from the question. This analysis was performed after the problems mentioned in Section 7.1 were dealt with.

The chapter is concluded by Section 7.5, on the translation of the questions for the multi-lingual tasks. The mistakes made due to the translation are analysed and it is made plausible the moment of translation (before or after the question is split) does not really influence the performance of the system.

7.1 Initial analysis

Because of the rash implementation of the system due to the CLEF deadline, the system turned out to have some bugs and some other problems not thought of. All these have been corrected before the real analysis was performed. The real 'bugs' are not mentioned here, but there were three somewhat bigger problems that should be explained.

- Type 3 questions do not need to contain a date or period, but they can. An example of such a question is *'Where did Reinhard Selten work, before he came to Bonn in 1984?'*. This kind of type 3 questions were recognized by the temporal QA system as being type 1 or 2, depending on the temporal expression in it. This was caused by a wrong sequence of checks. The sequence is changed and now it is first checked if it is a type 3 question and only when this seems not to be the case, it is checked if the question is of type 1 or 2.
- The Q-Object sent to retrieve a date for the restricting part of a type 3 question, often did not result in a date and it turned out it sometimes led to a date that was sure to be incorrect. The answer to the restricting part of a type 3 question that contains a direct temporal expression in itself, like the ones in the former bullet, can be checked against this

temporal expression. The example in the former bullet has restricting part *'before he came to Bonn in 1984'* and an answer to this restricting part, should at least be in 1984, but this was not always the case. This restriction is later built in. Also the restriction is built in that a found answer should contain a year, because the system is not able to construct a period or date from a date without year (because this has no meaning[29]). When a type 3 question that already contained a year is encountered and the answer to the restricting part is a date without a year, it is assumed the date belongs to the year mentioned in the question. So in the above mentioned question, when the first answer is *'19 November'*, the complete answer will be *'19 November 1984'*.

- The last problem was, as foreseen, that the lexicon with known nouns was too small. As explained in Section 5.2, the temporal part of a type 3 question, so a restriction by event, can come in the form of a PP in which the head of the NP has EVENT meaning. For example *'during the war'*, where *war* has EVENT meaning. To recognize these nouns, a lexicon with these nouns is needed. It was chosen not to construct such a lexicon, due to time constraints. As a result the lexicon used had only two nouns in it. To analyse if the idea works, the nouns should be known, so the needed nouns have been added afterwards. In the future a lexicon of some sort should be used.

7.2 CLEF 2005

The results on CLEF 2005, checked by assessors of CLEF, can be seen in Table 4. Notice that these are the results the system had before the problems and bugs in the last section were removed. For more information on CLEF see Section 2.2.

	Total	Factoid	Definition	Temporal
# Questions	200	120	50	30
# Cor. German–German	87	43	33	11
% Cor. German–German	44%	36%	66%	37%
# Cor. German–English	51	22	25	4
% Cor. German–English	26%	18%	50%	14%
# Cor. English–German	46	20	25	1
% Cor. English–German	24%	17%	50%	3%

Table 4 The results on CLEF[30]

[29] In the future, the timestamp of the newspaper could be used to give such a date meaning.

The system performs very well on definition questions, but those are not important here. The factoid questions are not interesting in the first instance, because in this thesis only the part of the system handling the temporally restricted factoid questions is described. However, the figures suggest a strong correlation between the factoid and temporally restricted questions. For German–German the percentages of correctly answered questions are respectively 36 and 37, for German–English 18 and 14 and for English–German 17 and 3. The 3% is a little too low for a clear correlation.

After the mistakes mentioned in the last section were removed, the system answered the questions again. This new run is looked at instead of the original run (for German–German), because mistakes and successes are not depending on the bugs and mistakes mentioned. The differences between the two runs can be seen in Table 5.

	CLEF	Afterwards
# Questions	30	30
# Correct answers returned	11 (37%)	8 (27%)
# *NIL* returned	10	9
# Correct *NIL* returned	4 (40%)	3 (33%)
# 'Real' answers returned	20	21
# Correct 'real' answers returned	7 (35%)	5 (24%)

Table 5 The results on the temporal questions during CLEF and afterwards

During CLEF, for German–German 11 questions were answered correctly. Of these eleven, four times the answer was *NIL* meaning no answer could be found in the corpus (and should not be found, since these answers were correct). The temporal QA system returned *NIL* 9 times. Afterwards, one correctly returned *NIL* has been changed to a 'real' but incorrect answer. The other nine returned *NIL* were also returned originally.

Respectively 20 and 21 'real' answers, so something different than *NIL*, were returned during CLEF and afterwards. Of these answers respectively seven and five were found to be correct. Five answers were correctly found during and after CLEF, two correct answers were not found anymore:

- For the question asking how many Chinese Airlines had deadly accidents in 1993, the problem was that *vier* should be extracted, but *zehn* was, from: *'Passagiers, die sich einer der immer zahlreicheren chinesischen Airlines anvertrauen, haben übrigens Grund zur Nervosität: unter den zehn Linienfluggesellschaften, die 1993 tödliche Unfälle verzeichneten, waren immerhin vier chinesische'*. So the QA system behind

[30] 'Cor.' means 'Correctly answered'. The number of factoid questions for German-English was 121 and the number of temporally restricted temporal questions was 29.

the temporal QA system just gave the wrong number the highest score.

- For the other question the problem is that only twenty sentences were extracted and the sentence with the right answer, simply was not one of these twenty.

In Table 6 it can be seen where the correct answers were found; directly, without splitting the temporal part from the 'real' question or after splitting. When *NIL* is returned, no answer was found before and after splitting.

	Total	Type 1: Date	Type 2: Period	Type 3: Event
Total correct answers	8	5	2	1
NIL correctly returned	3	2	1	0
Only directly found	2	2	0	0
Only found after splitting	1	0	1	0
Directly and after splitting	2	1	0	1

Table 6 The answers afterwards further specified

Two of the five 'real' answers returned were only found directly. These answers not being found after splitting was due to the maximum number of sentences retrieved from the document server. Because a lot of sentences are removed after the dates are checked, it would be better to retrieve more sentences for this task, so that recall would be higher. Two other answers were retrieved both before and after splitting. Only one answer would not have been found without splitting. This explains the strong correlation between the results on the factoid and the temporal questions. It also suggests splitting the questions is not very useful during CLEF. In the next section this is further elaborated on.

7.3 Overall result

As seen in the last section and Table 6, splitting the questions has not brought much to the number of correctly answered questions, although it should be recognized one extra correctly answered question on 30 questions answered is 3.33% of the questions. The question that arises is if splitting always leads to such low increase of performance. Therefore the corpus used is better looked at.

At first instance, the corpus seems to consist of questions a 'normal' user would ask the system. However, looking at the newspaper corpus the system uses to retrieve the answers from makes clear the question corpus is constructed using the newscorpus. Instead of thinking of some questions and then checking the corpus if

- The question *'Wer erhielt 1989 den Friedensnobelpreis?'* can easily be answered from:
  - *'Der Dalai Lama, der 1989 mit dem friedensnobelpreis ausgezeichnet wurde...'*
  - *'Der Dalai Lama, der 1989 den friedensnobelpreis erhalten hatte...'*

  This could still be accidentally.
- The question *'Wer war 1992 finnischer Außenminister?'* having the answer sentence *'Väyrynen selbst hatte als außenminister 1992...'* is already somewhat more suspicious.
- The question *'Wie viel Francs hat Mitterrand vor seiner Wahl 1981 als "Anwaltshonorare" bekommen?'* having answer sentence *'Laut einem Bericht des Richters hat auch mitterrand vor seiner wahl 1981 etwa 293 000 Francs bekommen - "anwaltshonorare", so der Präsident'* is too perfect to be an accident.

Because of this problem with the corpus, it seems likely in a 'normal' situation splitting would lead to a higher increase of performance. Moreover, even when only one extra question is answered correctly by applying the method proposed in this thesis, it is an increase, so the method is useful.

7.4 Splitting the questions

The overall performance of the system has been evaluated in the last sections. The focus of this thesis has been on question analysis, so in this section, one important part of the question analysis is elaborated on, the splitting of the questions in their 'main question' and the temporal restriction.

Every time the parser is mentioned, it is assumed the parser and POS-tagger are one. So when a word gets the wrong part-of-speech assigned, it is said the parser assigns the wrong part-of-speech. This because it is not really interesting here what causes this wrong tag. The named entity annotator is seen as a part on its own, because it is really clear what the output is of this part of the system, before it goes to the POS-tagger and parser.

In the next subsection the corpus used to analyse the results of the splitting is described, so that it is clear why this corpus is used. In Subsection 7.4.2 an overview is given of the results of the splitting. In Subsection 7.4.3 to 7.4.5, the results of respectively the type 1, 2 and 3 questions are further elaborated on.

In the final part of this section, in Subsection 7.4.6, it is evaluated how the extracted dates and periods of the type 1 and 2 questions are interpreted and internally saved.

7.4.1 Corpus

All the German questions on the CLEF website were put together, irrespective of the answer language. All the questions marked as temporal were extracted and duplicates removed, so that a total of 149 temporal questions resulted. Their further classification can be seen in Table 7. The questions in this corpus are thought of by a number of different people[31] and, most important, not by people who have something to do with the temporal QA system under development. Therefore, these questions seem to form a fair and objective corpus. The problem that the questions seem to be constructed from the answer corpus (see Section 7.3), is not a problem here because only the questions are of importance, not their answers.

Total	Type 1: Date	Type 2: Period	Type 3: Event	Not useful
149	78	24	43	4

Table 7 The from the CLEF corpus extracted temporal questions

The four questions that were not useful here were:

- *In welcher Klasse fuhr Mika Salo vier Jahre in Japan?* (*In which class has Mika Salo driven for four years in Japan?*)
- *Um wie viel haben sich die Olivenölimporte in die USA über die letzten 10 Jahre erhöht?* (*How much did the oliveoil imports in the USA rise during the last 10 years?*)
- *Wie viele Passagiere werden den Eurotunnel jedes Jahr durchqueren?* (*How many passengers will go through the Eurotunnel every year?*)
- *'Wo bildet sich jedes Jahr, während des Frühlings auf der Südhalbkugel, ein Loch in der Ozonschicht?'* (*'Where does each year, during spring on the Southern Hemisphere, form a hole in the ozone layer?'*)

The first question is not temporally restricted in the same way as the questions are 'usually' restricted in CLEF. The question is *'In welcher Klasse fuhr Mika Salo?'* and it is restricted in that the answer should be a *Klasse* in *Japan* and it should be a *Klasse* in which he raced for *vier Jahre*. This second restriction is temporal, but not a restriction by a time (or period) in history, like the other questions.

In the second question, the restriction by period is made relative to the current date. However, the temporal QA system has no knowledge of the

[31] For each language that is a possible target language in CLEF, there is a research group that thinks of questions answerable by the corpus in that language. These questions are translated into English and for each question language, these questions are translated from English into the question language. So the questions in German result from questions thought of by a large number of research groups.

current date, and looks for an explicit date in the question.

The third and fourth questions have somewhat the same 'problem' as the first question. The question is not restricted by a date (or period) in history. The difference with the first question is that there need not even be one answer here. Instead, a great number of answers is possible (for all the years to come) and the questioner wants the mean value of all these answers. For the fourth question it is most probable the hole will be at the same place every year (so there will only be one answer), but it need not be.

Although these kinds of questions should also be answerable by QA systems in the future, they do not really belong in the CLEF corpus of this year. A possible way to be able to give answers to the kind of questions like the second question, so with a relative date, was already mentioned in 7.1, namely by using the timestamp of the newspaper (when using a newspaper corpus).

7.4.2 Overview

In Table 8 it can be seen how well the questions are split. This means the number of questions is shown for which the temporal restriction is identified, and extracted from the question, like a human assessor would do. For the type 1 and 2 questions it is also shown if the extracted dates (type 1) or periods (type 2) are interpreted correctly. A date is correctly interpreted if the resulting internal representation is as expected by a human assessor and all the information available is used. So when the temporal part is *November 16, 1981*, the interpretation is incorrect if it is internally represented as = 1981:11. For a question restricted by event, the extracted temporal part can not be interpreted, an exact date for the temporal part first has to be found in the corpus.

	Total	Type 1: Date	Type 2: Period	Type 3: Event
# Questions	145	78	24	43
# Cor. split	113	73	19	21
% Cor. split	78%	94%	79%	49%
# Cor. interpreted	–	72	17	–
% Cor. interpreted (of cor. split)	–	99%	89%	–
% Cor. interpreted (of total)	–	92%	71%	–

Table 8 Splitting of the questions[32]

Although in Saquete et al. (2004b) it is shown how well their system performs on splitting temporal questions, the performance of the system implemented in the context of this thesis is not compared with their system. This is because they have not used the types distinguished in CLEF and therefore, their evaluation is done totally different from the evaluation performed here.

7.4.3 Splitting of type 1 questions

Of the five not correctly split type 1 questions, four are due to problems in deciding where prepositional phrases end. In *'Welche Kommission wurde von Major 1994 gegründet?'* (*'Which commission was founded by Major in 1994?'*), *'von Major 1994'* is extracted, because it is seen as a complete PP. In *'Welche Organisation kampierte auf der Castellana vor dem Winter 1994?'* (*'Which organization camped on the Castellana before the winter of 1994?'*), only *1994* is extracted because *'vor dem Winter 1994'* is not recognized as begin a PP. The difference between the two examples is that *Winter* has a 'temporal meaning' where *Major* does not. The parser should use this knowledge to decide where the PP ends. This is further elaborated on in Subsection 7.4.5.

In *'Wer war Ende 1994 Vorsitzender des Internationalen Bioethik-Komitees?'* (*'Who was the chair of the international bio-ethics committee at the end of 1994?'*) only 1994 is extracted. This is not the same problem as with the other four questions, because there is no PP here. SMES, the parser, recognizes *'Ende 1994'* is a noun phrase, with head *Ende* and modifier *1994*. This should be used by the temporal QA system.

The solution to this problem is to extract the complete NP when a year is in it. This will probably work for German, no counterexample has been thought of so far, but for English this will not work. Look for example at *'Who won the 2000 elections?'*. However, in English it is not possible to say something like *'Who was end of 1994 the chair of the international bio-ethics committee?'*. The temporal restriction has to be in a PP in English. The solution now is to distinguish between English and German (and possibly other languages) and using the approach of extracting a complete NP when a date is in it for German, but not changing the approach for English.

7.4.4 Splitting of type 2 questions

With the type 2 questions, there are some problems with LingPipe, the named entity annotator. In *'Wer wurde dreimal als bester Fußballspieler Europas in den Jahren 1971-1974 gewählt?'* (*'Who was chosen best soccer player of Europe three times in the years 1971-1974?'*) for example, only

[32] 'Cor.' is short for 'correctly'.

1974 is extracted, due to LingPipe splitting the period and just recognizing 1974 as a date. In two other examples LingPipe makes this mistake. A solution to this problem could be to search round a date in order to check if LingPipe did not miss a year. A better solution would probably be to retrain LingPipe with a corpus containing this kind of periods.

The fourth question has the same problem as mentioned with the type 1 questions: a prepositional phrase is extracted, where there should not be a PP. In Subsection 7.4.3 this was shortly mentioned and it is further elaborated on in Subsection 7.4.5.

The last incorrectly extracted temporal part is not really incorrect, but it has been assigned a wrong type by the system. The system has found a PP starting with *während* in *'Welche zwei Organisationen veranstalteten rivalisierende Snowboard-Ligen während der neunziger Jahre?'* (*'Which two organisations organized rival snowboard-leagues during the nineties?'*). Because it does not recognize *'neunziger Jahre'* as referring to a distinct period, it decides it has found an NP that leads to some date. The system would have recognized *'90er Jahre'*, but *'neunziger Jahre'* is not implemented. It can be argued this is not even wrong. There is a PP and you need world knowledge to know what is meant with this date. However, it seems useful to use a list with this kind of clear periods.

7.4.5 Splitting of type 3 questions

The temporal part of the 44 type 3 questions in the test corpus come in two broad categories, as shown in Section 5.2. The temporal restriction can come in the form of a subordinate clause or a prepositional phrase. The prepositional phrase can contain just one preposition, or it can be more complex, like *'during the attack on Iraq'*, which contains two prepositions. A subordinate clause can or can not contain a date, like *'before he became president in 1996'*. In this way there are four categories. In Table 9, it is shown how many questions of each type the corpus contains and how many were answered correctly.

	Total	Subord. excl. date	Subord. incl. date	PP simple	PP complex
# Questions	43	13	4	13	13
# Cor. split	21	9	3	9	0
% Cor. split	49%	69%	75%	69%	0%

Table 9 Splitting of type 3 questions[33]

It is clear the more complex prepositional phrases cause most problems. Of 13 examples, no one was extracted correctly. It was expected the subordinate clauses with dates in it would lead to mistakes due to the system recognizing the date instead of the subordinate clause. However, with 75% correct on this type and 69% correct on the subordinate clauses without date, the date does not seem to cause any trouble.

Subordinate clause without date

Of the temporal restrictions in the form of subordinate clauses without an explicit date, four questions went wrong. One of those questions did not even get to the temporal QA system, because the parser had problems with the apostrophes in the question.

In *'In wie viele Skandale war Tapie verwickelt, während er Chef in Marseille war?'* (*'In how many scandals was Tapie mixed up in, when he was director in Marseille?'*), *'während er'* is found to be a PP by the parser and is extracted as the temporal restriction. This mistake therefore is due to the parser. Another sentence fails because *Jahr(es)* is tagged as DATE by the named entity annotator and therefore extracted as the temporal part. This is also a problem the temporal QA system can not do anything about.

The fourth question is somewhat more complicated. In *'Wer dankte als Chef der CIA ab, als Aldrich Ames als Spion entlarvt wurde?'* (*'Who resigned as director of the CIA, when Aldrich Ames was unmasked as spy?'*) the word *als* returns three times. SMES, the parser, assigns this word every time the part-of-speech SUBORD, meaning this is a subordinating conjunctive. In such a case, the temporal QA system really can not decide which part is the subordinate clause. The only solution to this problem seems a parser that can decide which part exactly is the sought clause, so a parser that returns a parse of the sentence, at least defining which part the subordinate clause is.

Subordinate clause with date

The only question that went wrong in the category of type 3 questions with a subordinate clause containing a date, is the only one that does not really fit in this category. The problem in *'Wer wurde weiterhin mit verseuchten Blutprodukten beliefert, sechs Monate nachdem ein amerikanischer Blut-Screening-Test und ein Wärmebehandlungsprozess in anderen Ländern erhältlich waren?'* (*'Who was still provided with contaminated blood-products, six months after an American blood-screening-test and a warmth treatment process were available in other countries?'*) is that the relation is somewhat more complicated than usual. The relation is not just $>$, but something like $>\{6$ MONTHS$\}$ or 'six months greater than'. The system extracts just the

[33] 'Cor.' means 'correctly', 'Subord.' means 'temporal part is formed by a subordinate conjunctive' and 'PP' means 'temporal part is formed by a prepositional phrase'.

part starting with *nachdem* and therefore does not capture this more complex relation. Although there is only one example of this kind of question in the corpus, it is very interesting and could be included in the system in a next step. In such a case there should be sought for a 'relating date' near the conjunctive or preposition.

It is striking there is no example of a question in which the subordinate clause is in front or in the middle of the main clause. These questions would not be answered correctly by the temporal QA system as explained in Section 6.2. It seems this kind of questions are not natural in German or not natural to ask to a computer.

Prepositional phrase, simple

Of the four questions with a simple PP as temporal restriction that were not correctly split, one question did not get through the parser, although it seemed a 'normal' question. This is a problem the temporal QA system can not do anything about.

In two other questions, the temporal part was correctly extracted, but it was seen as a subordinate clause instead of a prepositional phrase. Both these PPs started with *während* (*during*), which can be both a preposition and a conjunctive. Because the parser can not decide the part-of-speech of *während* in these sentences, the word gets both parts-of-speech, but when it gets to the QA system, only the first possible part-of-speech is saved, because the implementation can only handle one part-of-speech. This one is the SUBORD, so a PP will never be found. Even when the implementation would let the two possible parts-of-speech exist, the temporal QA system would not be able to decide what to do: it needs an unambiguous parse.

There are also examples of the parser correctly deciding between *während* being a conjunctive or a preposition. In these cases the temporal QA system extracts the correct part of the sentence.

In the fourth question just a date was extracted, instead of a complete PP, this is the same problem as mentioned in Subsection 7.4.3 and 7.4.4 with splitting type 1 and 2 questions. This question shows it is difficult to decide, even for a human assessor, what is the correct temporal part to extract: in *'Wie viel Francs hat Mitterrand vor seiner Wahl 1981 als "Anwaltshonorare" bekommen?'* (*'How many francs did Mitterand receive as "lawyer fees" before his election in 1981?'*) it can be argued just *1981* should be extracted or *'vor seiner Wahl 1981'*. The last one seems to be the best choice since the *Wahl* further specifies the exact date in 1981 and *Wahl* has nothing to do with the 'real' question, the question that remains after extracting the restriction.

There is no syntactic difference between this example and the sentence *'Wie*

viel Francs hat Mitterrand vor seiner Wohnung 1981 als "Anwaltshonorare" bekommen?' (*'How many francs had Mitterand received as "lawyer fees" in front of his house in 1981?'*). It has to be known a *Wahl* has something to do with a moment in time and a *Wohnung* has not. This clearly shows a temporal QA system can not just use any parser, but needs a parser that uses some semantics.

Prepositional phrase, complex

The parser used does not try to form more complex PPs (by PP attachment). For the former use of this parser this was not necessary, but for temporal questions, especially these type 3 with complex PPs, but also the type 1 and 2 (as mentioned in Subsections 7.4.3 and 7.4.4), it is necessary to form complex PPs.

As shown above, at the end of the last paragraph, this is very difficult when parsing just on a syntactic level. The determination of the attachments of prepositions and subordinate conjunctions when parsing natural language is subject to ongoing research: look for example at Yeh and Vilain (1998) or Hartrumpf (1999), the last of which reports for German 88.6%–94.4% correct for binary attachment and 83.3%–92.5% for interpretation of ambiguities. Sometimes even a human assessor can not decide what is meant. In *'The man hits the dog with the stick'*, does the man carry the stick and use it to hit the dog? Or does the dog carry a stick and does the man hit him in some unspecified way? The problems with PP attachment have been the reason SMES does not do this attachment. See Neumann and Piskorski (2002) for more information.

In nine of the thirteen questions with more complex PPs, not the complete PP was extracted. In two cases the correct PP was extracted, but marked as a subordinate clause. This was due to the same problem as mentioned above with the simple PPs, so the parser not deciding on whether a word was a preposition or a conjunctive.

In one case, a subordinate clause was extracted, where a PP should have been extracted. This case is a little difficult: *'Wie alt war Nick Leeson zu der Zeit, als er zu Gefängnis verurteilt wurde?'* (*'How old was Nick Leeson at the time he was sent to prison?'*). Because of the *als* the temporal part is found to be a subordinate clause, but the complete part starting with *'zu der Zeit'* should be extracted as PP. *'zu der Zeit, als'* could be said to be a relation meaning =. In this case it should be in the lexicon with subordinating conjunctives in whole.

With the last wrongly identified complex PP, the problem was also the same as mentioned before with the simple PPs. To know, just a date was extracted, where a human assessor would extract a PP. However, in this case it is somewhat more complicated because there are really two tempo-

ral restrictions that syntactically do not belong together: in *'Gegen welchen Parteifreund trat Jacques Chirac 1995 während der Präsidentschaftskampagne in Frankreich an?'* (*'With which party friend did Jacques Chirac compete in 1995 during the presidential elections in France?'*) *1995* and *'während der Präsidentschaftskampagne in Frankreich'* do not syntactically belong together, but they do semantically. It would be strange just to call *1995* the temporal restriction. To handle this kind of restrictions, it is necessary for the temporal QA system to be able to deal with multiple restrictions, which it can not a the moment.

7.4.6 Interpretation

The interpretation of the correctly extracted dates and periods scores 99% and 89% correct, as seen in Table 8. The date and the two periods that are not interpreted correctly are:

- *Mitte der neunziger Jahre* (*in the middle of the nineties*)
- *am 11. Dezember* (*at December, 11*)
- *von 1992-1994.* (*from 1992-1994*)

The failure of interpreting the first two questions are due to the temporal QA system. In the first question, *90er* would have been recognized, but the system does not recognize *neunziger* at the moment (as mentioned before). In the second question the problem is that the date is not fully specified. Normally in such a question the questioner asks for the last *'11. Dezember'*. So if the current date is July 2005, the questioner probably asks for December 11, 2004. Depending on the context it could be different. The temporal QA system however has no knowledge of the current date or any other contextual knowledge, so it needs a date that at least contains a year. Of course this could be altered in the future, so that the system works as explained above. However, notice this would lead to an incorrect date chosen for the example above. In this case the date should be used as a 'normal' keyword and it should not be extracted. Because the system at the moment can not understand the date, it effectively works correctly: it first tries to find an answer with all original keywords, including the date, and then tries to extract the date. Because it can not extract this date, it simply returns the answer to the original question.

For the third question it could be argued the temporal part is not 100% perfectly extracted, because the period at the end should not be there. Now that it is there, the system does not recognize the part after the hyphen is a year and the result is > 1992. A solution would be to remove such a period if it is encountered, but it seems more logical to remove this period before, after named entity tagging.

7.5 Translation

The translation of the Q-Objects is not really good. Often important words are not translated. In the question *'Wer wurde dreimal als bester Fußballspieler Europas in den Jahren 1971-1974 gewählt?'* (*'Who was chosen best soccer player of Europe three times in the years 1971-1974?'*), of the ten keywords only six are translated, omitting *chosen*, *three times* and *soccer player* (and the '-' from *1971-1974*).

Translation of dates often leads to removal of the day or month. In *'Was geschah am 6. Juni 1944?'* (*'What happend on June 6th, 1944?'*), the date is translated as *'June 1944'*. It would probably not be too difficult to translate these dates using a short grammar and lexicon (English and German) instead of using online translators.

In the next subsection, the effects of the translation of Q-Objects on the splitting of the questions is evaluated. In the second subsection, it is evaluated if it would help to split questions first and translate them afterwards, instead of the current sequence of first translating and then splitting.

7.5.1 Effects on splitting

When answers to German questions were looked for in English documents, first a Q-Object is constructed, then this Q-Object is translated and only then the Q-Object is split in a restricting and a restricted part. Due to this translation seven questions are not split correctly anymore, three are split better now and one question was already split incorrectly and is now differently split incorrectly.

Of the seven questions that are not split correctly anymore, five times this was due to a period in the form of *'1990-2000'*, so two years with a dash in between and no spaces surrounding the dash. These periods were tagged as DATE by the named entity annotator, but they still caused problems during translation, so that they were not in the translated Q-Object. With a small hack in the part that does the translation, the dash was removed in such cases and it functioned perfectly.

In one question, the noun with EVENT meaning was not in the translated Q-Object and the part to split could not be found by the temporal QA system. In the seventh question, something goes wrong with the alignment of the translation, so that the conjunctive *dass* is replaced with *before*, which should have been the translation of the preposition *vor*. So instead of just a PP being extracted, a complete subordinate clause, which really is not a temporal restricting clause, is extracted.

In the tree questions from which a 'better' temporal restriction is extracted than before, the restriction is a PP starting with a preposition that can only

have temporal meaning, like *während* (*during*). The preposition is found and the PP is extracted. However, the head of the NP, so in *'während der Kandidatur für den Posten...'* (*'during the candidacy for the post...'*) *Kandidatur*, is not found in the translated Q-Object and as a result the complete sentence starting with *während* is extracted. Fortunately in these three cases this is the right thing to do.

The one question that leads to a different mistake than when searching in German answers was also a more complex PP, for which the head could not be found. In this case extracting the complete part starting with the preposition led to extracting too much.

7.5.2 Sequence of translation and splitting

In the current system, questions are translated and split afterwards. When questions would first be split and translated afterwards, the temporal restriction of a type 3 question needs to be rephrased so that it forms a correct sentence. This is not too difficult. The verb at the end of the sentence goes to the front and *Wann* is placed in front. If no verb is there *war* (*was*) or *waren* (*were*) is used instead.

Fifty questions have been split by hand and for the type 3 questions, the restricting part has been rephrased by hand. First only thirty questions were checked, but for the types 2 and 3, it was not clear then what the consequences were of the sequence of splitting and translation, so for these types the performance on ten additional questions (per type) were evaluated.

	Total	Type 1: Date	Type 2: Period	Type 3: Event
# Questions	50	10	20	20
# Better translated	9	0	3	6
% Better translated	18%	0%	15%	30%
# Worse translated	7	0	2	5
% Worse translated	14%	0%	10%	25%

Table 10 Questions split before translation instead of afterwards

As can be seen in Table 10, splitting the questions and translating them afterwards leads to a somewhat better translation: nine questions are split better than before and seven worse. For the types 2 and 3, this conclusion stays the same. Only for type 1 questions it really does not matter when they are being translated. However, there are a few things to notice:

- When the translation is somewhat better after or before splitting, the difference is mostly one keyword that is better or extra translated

(sometimes for example a verb like *starb* is translated as *did* before splitting and (correctly) as *died* afterwards).
- In this test, a perfect splitting is assumed. For the type 3 questions, this is not really realistic.
- The sentences were rewritten as correct sentences by hand. When the temporal QA system would do this, this could also lead to additional mistakes, resulting in a sentence that can not be translated (SMES, the parser can not handle the subordinate clauses on their one when they are not rewritten. If they are rewritten incorrectly, this could lead to SMES not parsing the sentence).

It seems translation after splitting will not lead to significantly better results.

8 Conclusions and recommendations

This chapter consists of two sections. In Section 8.1 conclusions are drawn from the work and findings described in this thesis. In Section 8.2, this is followed by recommendations for future work.

8.1 Conclusions

The first goal of the research reported in this thesis was to make clear what complex questions are, in order to be able to decide which type of complex questions would be chosen to handle in this thesis. In Chapter 2 and 3 the results are presented and it turned out complex questions have, as Saquete et al. (2004b, p. 568.) state it, have "... in common the necessity of an additional processing in order to be solved". After that, temporally restricted questions were chosen as the type of complex questions to handle in this thesis.

The main research question that resulted from this choice was, as stated in the first chapter:

> How can, in the context of open-domain multi-lingual question answering, temporally restricted questions be answered using an existing question answering system that was designed to answer only 'simple' factoid questions?

It was checked how other research groups handle complex questions and an approach was thought of to answer temporally restricted questions, making use of a 'normal' QA system, designed to handle 'simple' factoid questions (Quantico). This approach was implemented and put to work at CLEF 2005.

It turned out for CLEF the results were almost as good as the results on the 'simple' factoid questions. However, this was partly caused by the way the questions for CLEF were constructed. A lot of answers for questions could be found using only one sentence of the corpus. This led to finding all but one answer (eight correct answers were found for thirty questions) for the German–German task without splitting the questions into the 'real' question and the 'temporal restriction'. Effectively, these answers were found using the 'simple' approach.

Although this result may not seem to support the chosen approach, it was clarified in Section 7.3 that when the questions would not be constructed from the corpus, but would be thought of by 'normal' users, the answers would more often be not directly found in the corpus and splitting would be necessary.

For questions with a restriction by DATE, splitting will often be not really

useful. When asking which painter died in 1939, the date can be stripped off, but if a possible answer should be checked, there has to be a date and it will only be found to be correct if the date is 1939. When a question asks for all painters born before 1800, sentences containing the correct answers explicitly will not contain the year 1800, so splitting is needed. For restrictions by PERIOD the reasoning is analogous. For restrictions by EVENT splitting will often help, but in some cases it will not. For example in *'Who was president during the Second World War?'* it is very probable a sentence in the corpus can be found telling who it was, so splitting is not necessary.

The splitting part of the temporal QA system was tested on its own and it functioned really well for questions restricted by DATE (94% correct) and PERIOD (79% correct). For questions restricted by EVENT only 49% was split correctly. However, this was mainly due to the parser not doing PP attachment. When the cases where PP attachment was needed are not taken into account, 70% of the questions restricted by EVENT would be correctly split.

Except for the PP attachment, which is really needed, it seems it suffices to use a parser that does not return a full, deep parse.

The goals of this thesis, namely to clarify what complex questions are and to design, implement and test a system that could answer one type of these complex questions, were clearly met.

8.2 Recommendations

Although the approach seems useful, this could not really be shown during CLEF. Therefore, it would be useful to get a larger corpus of questions, 200 questions for example, not constructed from the answer corpus. With such a corpus it could be really tested what the increase of correctly answered questions is when the questions are split. Because of a lack of time, this has not been done yet.

A parser that does PP attachment should be integrated in the system or the temporal QA system should do the PP attachment itself. Probably the questions that were restricted by EVENT in the form of a more complex PP, so a PP containing more than one preposition, would be split correctly then, but of course this can only be known for sure when it is tested.

As explained in Section 7.1, a lexicon is needed of nouns with EVENT meaning to disambiguate the meaning of PPs when the preposition has multiple meanings.

Besides these three main recommendations, there are some other recommendations:

- When answer sentences are checked for being temporally correct in relation to the question, it is now only checked if any date in the answer sentence satisfies the temporal restriction in the question. It is not checked if the possible answer and the date are related to each other. Especially in a newspaper corpus, there are long sentences having possibly multiple conjunctives, so that it can happen a date and a possible answer are in the same sentence, but are not related to each other. Therefore, answer sentences should be analysed.

- Analogous to the last point, sentences could be syntactically checked. As explained in Subsection 5.1.2, there is a difference between *'Juliana was queen before Beatrix'* and *'Wilhelmina was queen before Juliana'* when searching for an answer for *'Who was queen before Juliana?'*.

- As mentioned in Section 5.2, conjunctives with multiple meanings cause problems in deciding if the subordinate clause they start have temporal meaning. It was explained that probably for each ambiguous conjunctive different checks should be implemented to check which meaning is used. This should be worked out.

- Intensionally specified or underspecified temporal expressions in question and answer sentences could be used, for example:
  - *ten days ago*
  - *last Friday*
  - *December, 17th*

  Especially in a newspaper corpus these are often found, but at the moment the temporal QA system can not use them. If the system can use the date of the article and a calendar, it could construct 'real' dates from these expressions.

- Related to the last bullet is the point mentioned in Subsection 7.4.5: in *'Who was still provided with contaminated blood-products, six months after an American blood-screening-test and a warmth treatment process were available in other countries?* the relation is not just $>$, but something like $>\{\text{SIX MONTHS}\}$. Such relations can not be handled at the moment, but are interesting for a future system.

- Answers could be checked, a little like Prager et al. (2003, 2004) have done. When for example a question was *'Who was president in 1980?'* and two answers are received, for example *Carter* and *Reagan*, the question could be rephrased using the answers, so that two questions are constructed: *'When was Carter president?'* and *'When was Reagan president?'*. Hopefully for the first question the answer *'1977 – 1981'* is received and for the second *'1981 – 1989'*. In this case it is known that *Carter* is the correct answer.

- For terms like *summer*, *Middle Ages* and *eighties* a lexicon could be used that specifies which period is meant, as mentioned in Subsection 7.4.4. However, the problem with for example *summer* or *Middle Ages* is, it is not fully specified what is meant, because different ideas about this exist. A solution for this should be thought of. For *eighties*, this is no problem.

- Sometimes, a relation or date is clarified by a verb or noun used. For example: *'Who was queen when World War II ended?'* or *'Who was queen at the end of World War II?'*. When the verb and noun *end* would not be there, the temporal part would have a PERIOD equivalent, namely = 1939 – 1945. However, now the equivalent is just a DATE, = 1945. These verbs and nouns are not used yet.

- When a type 1 or 2 question is stripped of its temporal restriction, a query is constructed from the remaining 'real question'. To this query, the restriction is added that at least a date should be in the possible answer sentences, so that these answers can be checked against the original temporal restriction. This is all fully explained in Section 5.1. In Section 7.2 it was shown that it sometimes happened that none of the retrieved results for such a query contained a correct date, so all sentences were rejected and no answer was found, although the correct answer was in the corpus. This is due to the maximum number of sentences retrieved from the document server, Lucene. It seems useful to retrieve more sentences in such a case. When for example no correct answer is found in the first twenty sentences retrieved, another twenty should be retrieved and so on till an answer is found or no possible answer sentences can be retrieved anymore.

References

Allen, J. F. (1983a). Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery*, *26*(11), 832–843.

Allen, J. F. (1983b). Recognizing intentions from natural language utterances. In M. Brady & R. Berwick (Eds.), *Computational models of discourse* (pp. 107–166). Cambridge, Massachusetts: MIT Press.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence*, *23*(2), 123–154.

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D. et al. (2000). *Issues, tasks and program structures to roadmap research in question & answering (Q&A).* Retrieved on January 20, 2005, from http://www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc

Carbonell, J., Harman, D., Hovy, E., Maiorano, S., Prange, J. & Sparck-Jones, K. (2000). *Vision statement to guide research in question & answering (Q&A) and text summarization.* Retrieved on January 22, 2005, from http://www-nlpir.nist.gov/projects/duc/papers/Final-Vision-Paper-v1a.pdf

CLEF. (2004). *Cross-language evaluation forum 2004. evaluation of question answering systems for spanish at CLEF-2004. pilot task.* Retrieved on February 15, 2005, from http://terral.lsi.uned.es/QA/2004/pilot2004/esQApilot-translated.pdf

Diekema, A., Yilmazel, O., Chen, J., Harwell, S., He, L. & Liddy, E. (2003). What do you mean? finding answers to complex questions. In *Proceedings of the AAAI spring symposium: New directions in question answering* (pp. 87–93).

Graesser, A. C., Person, N. & Huber, J. (1992). Mechanisms that generate questions. In T. W. Lauer, E. Peacock & A. C. Graesser (Eds.), *Questions and information systems* (chap. 9). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Green, C. & Raphael, B. (1968). The use of theorem proving techniques in question-answering systems. In R. Blue (Ed.), *Proceedings of the 23rd national conference of the association of computing machinery* (pp. 169–181). Princeton, New York: Brandon Systems Press.

Green, P., Wolf, A., Chomsky, C. & Laugherty, K. (1963). BASEBALL: An automatic question answerer. In E. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York, New York: McGraw-Hill.

Harabagiu, S., Maiorano, S., Moschitti, A. & Bejan, C. (2004). Intentions, implicatures and processing of complex questions. In S. Harabagiu & F. Lacatusu (Eds.), *Proceedings of the workshop on pragmatics of question answering at HLT-NAACL 2004* (pp. 31–42).

Hartrumpf, S. (1999). Hybrid disambiguation of prepositional phrase attachment and interpretation. In *Proceedings of the joint conference on empirical methods in natural language processing and very large corpora (EMNLP/VLC-99)* (pp. 111–120).

Krause, J. (1982). *Mensch-maschine-interaktion in natürlicher sprache: Evaluierungsstudien zu praxisorientierten frage-antwort-systemen und ihre methodik.* Tübbingen, Germany.

Lang, K., Graesser, A. & Hemphill, D. (1990). The role of questioning in knowledge engineering and the interface of expert systems. *Poetics*, *19*(1–2), 143–166.

Lehnert, W. (1978). *The process of question answering.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Magnini, B., Vallin, A., Ayache, C., Erbach, G., Peñas, A., Rijke, M. de et al. (2004). Overview of the CLEF 2004 multilingual question answering track. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck & B. Magnini (Eds.), *Working notes for the CLEF 2004 workshop* (pp. 281–294).

Mani, I. & Wilson, G. I. (2000). Robust temporal processing of news. In *Proceedings of the 38th annual meeting of the association for computational linguistics* (pp. 69–76). San Francisco, California: Morgan Kaufmann.

Manning, C. (2004). *(TREC-style) question answering systems* [Lecture Slides]. Retrieved on February 9, 2005, from http://www.stanford.edu/class/cs224n/new_handouts/cs224n-QA.pdf

Moldovan, D., Harabagiu, S., Clark, C., Bowden, M., Lehmann, J. & Williams, J. (2004). Experiments and analysis of LCC's two QA systems over TREC 2004. In E. Voorhees & L. P. Buckland (Eds.), *TREC 2004 conference note book* (pp. 21–30). Gaithersburg, Maryland: National Institute of Standards and Technology.

Monz, C. (2003). *From document retrieval to question answering.* Amsterdam, Netherlands: Universiteit van Amsterdam.

Neumann, G. & Piskorski, J. (2002). A shallow text processing core engine. *Journal of Computational Intelligence*, *18*(3), 451–476.

Neumann, G. & Sacaleanu, B. (2005). Experiments on robust NL question interpretation and multi-layered document annotation for a cross-language question/answering system. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck & B. Magnini (Eds.), *Proceedings of CLEF 2004* (pp. 411–422). Berlin, Germany: Springer.

Pérez-Coutiño, M., Solorio, T., Montes-y-Gómez, M., López-López, A. & Villaseñor-Pineda, L. (2004). The use of lexical context in question answering for spanish. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck & B. Magnini (Eds.), *Working notes for the CLEF 2004 workshop.* Retrieved on June 10, 2005, from http://clef.iei.pi.cnr.it/2004/working_notes/WorkingNotes2004/46.pdf

Prager, J., Chu-Carroll, J. & Czuba, K. (2004). Question answering using constraint satisfaction: QA-by-dossier-with-contraints. In D. Scott, W. Daelemans & M. Walke (Eds.), *Proceedings of the 42nd annual meeting of the association for computational linguistics: ACL 2004* (pp. 574–581).

Prager, J., Chu-Carroll, J., Czuba, K., Welty, C., Ittycheriah, A. & Mahindru, R. (2003). IBM's PIQUANT in TREC 2003. In E. Voorhees & L. Buckland (Eds.), *The twelfth text retrieval conference (TREC 2003). NIST special publication 500-255* (pp. 283–293). Gaithersburg, Maryland: National Institute of Standards and Technology.

Pustejovsky, J., Ingria, R., Saurí, R., Castaño, J., Littman, J., Gaizauskas, R. et al. (2005). The specification language TimeML. In I. Mani, J. Pustejovsky & R. Gaizauskas (Eds.), *The language of time: A reader.* Oxford, Great Britain: Oxford University Press.

Raphael, B. (1968). SIR: A computer program for semantic information retrieval. In M. L. Minsky (Ed.), *Semantic information processing* (pp. 33–134). Cambridge, Massachusetts: MIT Press.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery*, *12*(1), 23–41.

Saquete, E., Martínez-Barco, P., Muñoz, R. & Vicedo, J. (2004a). Evaluation of complex temporal questions in CLEF-QA. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck & B. Magnini (Eds.), *Working notes for the CLEF 2004 workshop.* Retrieved on June 10, 2005, from http://clef.iei.pi.cnr.it/2004/working_notes/WorkingNotes2004/54.pdf

Saquete, E., Martínez-Barco, P., Muñoz, R. & Vicedo, J. (2004b). Splitting complex temporal questions for question answering systems. In D. Scott, W. Daelemans & M. Walker (Eds.), *Proceedings of the 42nd annual meeting of the association for computational linguistics: ACL 2004* (pp. 566–573).

Schilder, F. & Habel, C. (2001). From temporal expressions to temporal information: semantic tagging of news messages. In *Proceedings of the ACL 2001 workshop on temporal and spatial information processing* (pp. 65–72).

Simmons, R. F. (1965). Answering english questions by computer: A survey. *Communications of the Association for Computing Machinery*, *8*(1), 53–70.

Simmons, R. F. (1966). Storage and retrieval of aspects of meaning in directed graph structures. *Communications of the Association for Computing Machinery*, *9*(3), 211–215.

Simmons, R. F. (1969). Natural language question answering systems: 1969. *Communications of the Association for Computing Machinery*, *13*(1), 15–30.

Tanev, H., Negri, M., Magnini, B. & Kouylekov, M. (2004). The DIOGENE question answering system at CLEF-2004. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck & B. Magnini (Eds.), *Working notes for the CLEF 2004 workshop.* Retrieved on June 10, 2005, from http://clef.iei.pi.cnr.it/2004/working_notes/WorkingNotes2004/40.pdf

Thompson, F. (1966). English for the computer. In *Proceedings of the fall joint computer conference* (pp. 349–356). New York, New York: Spartan.

Voorhees, E. (2002). Overview of the TREC 2002 question answering track. In E. Voorhees & L. P. Buckland (Eds.), *The eleventh text retrieval conference (TREC 2002). NIST special publication 500–251* (pp. 57–68). Gaithersburg, Maryland: National Institute of Standards and Technology.

Voorhees, E. (2003). Overview of the TREC 2003 question answering track. In E. Voorhees & L. P. Buckland (Eds.), *The twelfth text retrieval conference (TREC 2003). NIST special publication 500–255* (pp. 54–68). Gaithersburg, Maryland: National Institute of Standards and Technology.

Voorhees, E. (2004). Overview of the TREC 2004 question answering track. In E. Voorhees & L. P. Buckland (Eds.), *The thirteenth text retrieval conference proceedings (TREC 2004). NIST special publication 500–261.* Gaithersburg, Maryland: National Institute of Standards and Technology. Retrieved on June 20, 2005, from http://trec.nist.gov/pubs/trec13/papers/QA.OVERVIEW.pdf

Warren, D. & Pereira, F. (1982). An efficient easily adaptable system for interpreting natural language queries. *Computational Linguistics*, *8*(3–4), 110-122.

Weizenbaum, J. (1966). ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery*, *9*(1), 36–45.

Wenger, E. (1987). *Artificial intelligence and tutoring systems.* Los Altos, California: Morgan Kaufmann.

Winograd, T. (1972). *Understanding natural language.* New York, New York: Academic Press.

Woods, W. (1968). Procedural semantics for a question-answer machine. In *Proceedings of the fall joint computer conference* (pp. 457–471). New York, New York: Spartan.

Yeh, A. S. & Vilain, M. B. (1998). Some properties of preposition and subordinate conjunction attachments. In *Proceedings of COLING-ACL '98: 36th annual meeting of the association for computational linguistics and 17th international conference on computational linguistics* (Vol. 2, pp. 1436–1442). San Francisco, California: Morgan Kaufmann.