Series of Publications A
Report A-2007-3

Methods for Answer Extraction in Textual Question Answering

Lili Aunimo

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium XIV, University Main Building, on June 12th, 2007, at noon.

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

ISSN 1238-8645
ISBN 978-952-10-3992-8 (paperback)
ISBN 978-952-10-3993-5 (PDF)
Computing Reviews (1998) Classification: H.3.3, H.3.4, I.2.1, I.5.4
Helsinki 2007
Helsinki University Printing House

Methods for Answer Extraction in Textual Question Answering

Lili Aunimo

Department of Computer Science
P.O. Box 68, FI-00014 University of Helsinki, Finland
lili.aunimo@iki.fi

PhD Thesis, Series of Publications A, Report A-2007-3
Helsinki, June 2007, 127 + 19 pages
ISSN 1238-8645
ISBN 978-952-10-3992-8 (paperback)
ISBN 978-952-10-3993-5 (PDF)

Abstract

In this thesis we present and evaluate two pattern matching based methods for answer extraction in textual question answering systems. A textual question answering system is a system that seeks answers to natural language questions from unstructured text. Textual question answering systems are an important research problem because as the amount of natural language text in digital format grows all the time, the need for novel methods for pinpointing important knowledge from the vast textual databases becomes more and more urgent. In addition to this, textual question answering systems form a well-defined framework with lots of existing evaluation data in which new methods can be developed and evaluated. The separate subproblem of developing answer extraction methods for extracting answers from unstructured text is an interesting problem not only by itself but also because it is quite similar to the problems of information extraction from text and of semantic annotation of text. Thus, answer extraction methods may be generalized for these problems also. In this thesis, we concentrate on developing methods for the automatic creation of answer extraction patterns. A new type of extraction patterns is developed as well. The pattern matching based approach chosen is interesting because of its language and application independence.

The answer extraction methods are developed in the framework of our own question answering system. Publicly available datasets in English are used as training and evaluation data for the methods. The techniques developed

are based on the well known methods of sequence alignment and hierarchical clustering. The similarity metric used is based on edit distance.

The new answer extraction patterns developed consist of the most important words in the question, part-of-speech tags, plain words, punctuation marks and capitalization patterns. The two new methods for creating answer extraction patterns are called the concatenation based method and the alignment based method. The performance of the answer extraction patterns and of the methods for generating them is measured indirectly through the performance of the patterns in the answer extraction task of a question answering system. The difference in performance between the concatenation based and the alignment based answer extraction pattern generation methods is not significant when evaluated using the evaluation data. However, when evaluated using the training data and when taking into account only the first answer candidate, the alignment based method performs significantly better than the concatenation based one. The average accuracy of the question answering system when evaluated with evaluation data is about 0.17.

The main conclusions of the research are that answer extraction patterns consisting of the most important words of the question, plain words, part-of-speech tags, punctuation marks and capitalization patterns can be used in the answer extraction module of a question answering system. This type of patterns and the two new methods for generating answer extraction patterns provide average results when compared to those produced by other systems using the same dataset. However, most answer extraction methods in the question answering systems tested with the same dataset are both hand crafted and based on a system-specific and fine-grained question classification. The significance of the results obtained in this thesis reside in the fact that the new methods require no manual creation of answer extraction patterns. As a source of knowledge, they only require a dataset of sample questions and answers, as well as a set of text documents that contain answers to most of the questions. The question classification used in the experiments is a standard one and does not require additional work as it is provided by the evaluation data.

Computing Reviews (1998) Categories and Subject Descriptors:

H.3.3 **Information Storage and Retrieval**: Information Search and Retrieval

H.3.4 **Information Storage and Retrieval**: Systems and Software
I.2.1 **Artificial Intelligence**: Applications and Expert Systems - Natural language interfaces
I.5.4 **Pattern Recognition**: Applications - Text processing

General Terms:
Algorithms, Experimentation, Information Systems

Additional Key Words and Phrases:
Question answering systems, evaluation, hierarchical clustering, edit distance, vector space model

Acknowledgements

I am most grateful to my supervisor Helena Ahonen-Myka for her tireless guidance, patience and encouragement throughout my studies and throughout the entire process of writing this thesis. I am also very grateful to my other supervisor Greger Lindén for his insightful questions and comments on the thesis manuscript. I also wish to thank Walter Daelemans and Jussi Karlgren for reviewing the manuscript and for giving helpful comments on how to improve it.

I acknowledge the Department of Computer Science of the University of Helsinki for providing me with excellent working conditions. I am especially grateful to the computing facilities staff of the department for ensuring the fluent operation of the computing environment.

I have received financial support for my PhD studies from the Graduate School of Language Technology in Finland (KIT) and from the From Data to Knowledge (FDK) research unit. I gratefully acknowledge this support.

Many co-students both at the Department of Computer Science and at the KIT Graduate School have provided me invaluable intellectual and emotional support. I also wish to thank my colleagues at Evtek for bearing with me when I have been stressed because of an unfinished thesis.

I am most indebted to my parents for their support. I am also grateful to my sister and all my friends for your caring and for distracting me from work. Last, but foremost, I wish to thank my daughter and my husband for their love. Especially the encouragement of my husband Tomi has been vital for this work.

Contents

Chapter 1

Introduction

This introductory chapter first explains what question answering and answer extraction are all about. Secondly, the field of textual question answering is given a closer look. The third section lists the contributions of the thesis and describes its organization.

1.1 Question answering and answer extraction

Question answering (QA) is the problem of delivering precise answers to natural language questions [HG01, HM03]. This is an old problem in artificial intelligence research where it has been studied as a part of expert systems and natural language user interfaces since the early 1960s [GCL61, Sim65, Woo73]. Today, most research on QA is done in the Information Retrieval (IR) community [Mon03, AS05], but there is also research on QA systems in the natural language processing, artificial intelligence and user interface communities.

QA is a technology that takes text retrieval beyond search engines by pinpointing answers instead of delivering ranked lists of documents. Much of the effort lies in answering wh-questions, i.e. questions beginning with *who*, *what*, *where*, *why*, *which*, *when*, and *how*, and extracting single facts, lists of facts, or definitions from large corpora of text documents. The QA tracks at evaluation forums such as TREC[1] and at its European and Japanese counterparts CLEF[2] and NTCIR[3] have a major role in directing the research. This kind of QA is also called textual QA [HM03], and it is

[1] Text REtrieval Conference, http://trec.nist.gov/

[2] Cross Language Evaluation Forum, http://www.clef-campaign.org/

[3] NTCIR (NII Test Collection for IR Systems) Project, http://research.nii.ac.jp/ ntcadm/index-en.html

exactly the kind of task that is tackled by the new methods presented in this thesis. Textual QA will be described in more detail in the next section.

Answer extraction is the act of extracting text strings constituting the exact answer or answers to a question from a text snippet. The text snippet from which the answer is extracted may vary in size: it may consist of only a title or of an entire text document. The text snippet is typically retrieved using information retrieval techniques. The query words are usually formed by extracting some words from the natural language question using natural language processing techniques or some heuristics, and by adding some new words using different query expansion techniques. There are several different techniques for performing answer extraction in QA. The two main categories into which these techniques may be classified are the one based on pattern matching and the one based on logical inference and proofs. The methods based on pattern matching may themselves be categorized according to the different types of preprocessing they require (for example, syntactic parsing and named entity recognition) and according to the way the patterns are formed, i.e. manually or automatically. The methods based on logical inference and proofs form a small minority among answer extraction methods and they are dealt with only very briefly in this thesis. Existing answer extraction methods are described in more detail in Section 3. The methods presented and evaluated in this thesis are based on pattern matching. A major challenge for answer extraction methods based on pattern matching is the creation of answer patterns. The reason for this is that the number of patterns and the amount of detail in them are often very high and thus producing them manually is very time consuming. In this thesis, two methods for the automatic generation of answer extraction patterns are described and evaluated.

Question answering and answer extraction are closely related to the problem of information extraction (IE) from text. IE is the act of extracting facts from text to fill a predefined template [GW98]. This template may be regarded as a database schema and the extracted facts a record in the database. An example of a template could be *movie*, *director*, *main actor*, *duration* and *year of premiere*. Now, the IE task would consist of finding strings from text to fill in the fields of the table. An IE system may be regarded as a QA system that has a predefined set of questions that it is able to answer, i.e. those specified by the template. However, as one of the major challenges in IE system research has been to develop systems that are easily portable from one domain (i.e. template) to another, the task is very similar to open domain QA, where the system may answer questions belonging to any domain. One special technique used in QA that

is especially close to IE is the technique of *answering questions before they are asked* [FHE03, JdRM04]. When using this technique, the answers to frequently asked questions and to frequently occurring question types are extracted from text in a preprocessing phase before the system is taken into use or always when it is updated. Currently, many of the best performing IE systems rely on pattern matching techniques [SS05].

1.2 Textual question answering

Textual QA systems are systems that extract the answer to a question from a plain text document collection. The motivation for developing textual QA systems is twofold. Firstly, there is an increasing need for intuitive user interfaces through which the ever growing amounts of unstructured, plain text data can be accessed. This need has increased along with the development of the World Wide Web (WWW) which has made information systems available also to non-expert users. The second reason for the need of textual QA systems is that the information overload with which users are faced demands for new systems that help in finding the relevant data in a more efficient, accurate and user-friendly manner.

Textual QA systems are typically composed of the question analysis, information retrieval and answer extraction components. The answer extraction component is often quite complex. It may involve natural language analysis, logical inference and/or it might involve pattern matching, in which case it requires large sets of patterns. The novel methods described in this thesis are pattern based. In addition to defining a new kind of pattern, an important part of the method is a technique for automatically generating the patterns. This is because the number and nature of the patterns is such that hand crafting them would have been an error prone and tedious endeavor due to the amount of the patterns – about 80000 – and due to the amount of detail in them.

In a textual QA system, the answer to a question can appear anywhere in the textual part of a document collection: in paragraphs, titles, captions, and so on. It can also be scattered in various places of the document collection. This is often the case for questions asking for a list. For example, *Name the major rivers of Europe.* Answering a question may also require making inferences. For example: the question *Who is the current CEO of Company X?* might require making inferences from the publication dates of the text material from which the answer is searched and from the date when the question is asked. Although performing temporal inferences has been studied in QA [Voo04, VMG+06] and although it is an important part

of a QA system, it is out of the scope of the work presented here. In the problem setting of this thesis, it is assumed that an answer is composed of a continuous single text snippet and that the evidence justifying it is presented as plain text in the textual part of the document surrounding the text snippet.

Another general issue in textual QA that needs to be defined is the size of the answer. According to the guidelines of the current evaluation campaigns for textual QA [CLE06, VD05, SChCL05], only the contents of the answer with regard to the question is decisive and thus the length of the answer may vary from a single word to several sentences. According to the guidelines of the evaluation campaigns, the answer should contain the necessary information and no superfluous information. Thus, for a very general and open-ended question such as: *What is the history of the World's languages?*, the answer could in principle be several documents. For some well defined questions such as *What is the country code for Finland?*, the answer is typically very short. The general focus in research on QA systems has been in processing only questions soliciting relatively short and precise answers. This applies even to the last question in information nuggets (i.e. questions of type *Other*) and to questions of type *List* although the answer may contain quite a number of words.(Information nuggets and different question types such as *Other* and *List* will be explained in the next chapter.)The answers are still short and precise because they consist of individual pieces that have been collected from various parts of the document collection. For instance, in an information nugget, each separate piece of information can be considered as a short answer to a single and very precise question. For example, the information nugget whose target is *Tarja Halonen?* can be broken down into several precise and short questions such as: *When was Tarja Halonen born?* and *What is the profession of Tarja Halonen?*. The proper granularity of an answer to a natural language question has been studied at the natural language query track of the INEX[4] Initiative, see e.g. [WG04, GW06]. However, in this work we present methods for only dealing with relatively short answers to factoid and definition questions. This has also been the focus of recent QA research [LK05].

To be precise, an answer that our system retrieves may be of three different types: 1) a named entity such as a location or a name of an organization, as defined in the MUC-7 Named Entity Task Definition [CR97], 2) an entire sentence or a part of it containing a definition of a person or of an organization, or 3) an entity other than those defined in MUC-7, such

[4]Initiative for the Evaluation of XML Retrieval, http://inex.is.informatik.uni-duisburg.de:2004/

as the name of a film, a verb phrase, an artifact or a nationality. In general terms, the answer may be any sequence of words that is not longer than one sentence.

1.3 Contributions and organization of the thesis

The main research questions addressed are:

1. What kind of information should we include in an answer extraction pattern? Should it contain morphological and/or syntactic information? Should it contain information extracted from the question?

2. How do we transform training data i.e. questions and the corresponding answers and their contexts into answer extraction patterns?

3. How do we apply the answer extraction patterns, i.e. how to map a new question into a set of answer extraction patterns and execute them?

4. How do we score the extracted answers in order to be able to choose the best one?

The above research questions are either novel or no definitive answer has yet been found to them despite various attempts. Some answers to the first, third and fourth research problems have been published (see e.g. [RH02], [XWL04], [FHE03] and [KL03]), but no claims of having found the best or even some recommended ways to solve the problems have been made. The types of answer extraction patterns already suggested and evaluated as a part of a QA system, existing ways of applying answer extraction patterns in a QA system and existing ways of scoring extracted answer candidates are described in Section 3.1. The second research question concerning the method in which answer extraction patterns can be induced from training data is a relatively novel problem. Prior to the work at hand, most answer extraction patterns have been either crafted manually or information extraction pattern generation methods have been used. This prior work is also presented in more detail in Section 3.1. All of the four research questions are important because they form the core of the answer extraction component of a textual QA system.

This thesis provides answers to the four research questions. In addition to this, the methods developed in this thesis are also applicable in other contexts than in QA. They can be used to induce information extraction patterns or patterns for the semantic annotation of text, among others. In

more general terms, the methods can be used to produce patterns for any extraction or annotation task where the input is preprocessable into the same quite general format required by the method and where the amount of training data is more or less equal to the amount used in the experiments presented in this thesis.

The thesis is organized as follows. *Chapter 2* presents the general framework of textual question answering systems. It introduces the central concepts and the general architecture used as well as the evaluation campaigns and measures. *Chapter 3* gives an overview of the state of the art in answer extraction. This chapter is mostly about pattern matching based methods for answer extraction. It describes different types of answer extraction patterns, methods for creating them and finally some results concerning their performance. *Chapter 4* presents the novel QA system along with the data that is used to generate the answer extraction patterns and that is used in the evaluation of the methods presented in this thesis.

Chapter 5 describes the techniques that are used in the novel QA system and in the generation of the answer extraction patterns. These techniques include the vector space model, hierarchical clustering, edit distance and alignment. The QA system uses the vector space based model of information retrieval in order to find document candidates from which text snippets are extracted. When creating the answer extraction patterns, similar preprocessed text snippets are grouped together using agglomerative hierarchical clustering. Edit distance is used to measure the distance between preprocessed text snippets and multiple string alignment is used to form regular expressions from a set of preprocessed text snippets.

Chapter 6 describes the novel answer extraction patterns and how they can be generated from training data. Two separate methods for generating them have been devised. They are the concatenation based and the alignment based methods. The chapter also presents how the answer extraction patterns are used in a QA system and how the answer candidates extracted using the patterns are scored.

Chapter 7 describes the experimental setting that is used to evaluate the novel answer extraction methods. Results of experiments concerning the concatenation and alignment based methods are given separately. The experiments are conducted using both the training data and a previously unseen test data set. *Chapter 8* presents an analysis of the results given in the previous chapter. It also compares the novel method and the results obtained in the experiments with the state of the art. Finally, some limitations and ideas for future work are given. *Chapter 9* concludes the thesis.

Chapter 2

General framework of textual question answering

This chapter introduces the general framework of question answering (QA). Special emphasis is put on textual QA, which is both the focus of most research in the field nowadays as well as the focus of this thesis. First, QA systems are described by both defining the central concepts related to them and by describing the general architecture of a QA system. Subsequently, the evaluation of textual QA systems is described by introducing several initiatives for the evaluation of QA systems and by defining the measures that are commonly used for assessing their performance.

2.1 Textual question answering systems

This section introduces the central concepts and terms related to textual QA systems and introduces a general system architecture of a textual QA system.

2.1.1 Central concepts

Textual QA systems are information systems that receive as input a natural language question, search for the answer from a large database of unstructured text and finally return a text string containing the exact answer to the question. Just to name a few examples, the large database of unstructured text may consist of newspaper text, of user manuals concerning the products of a company or of documents in the WWW. Textual QA typically combines methods from the fields of information retrieval and natural language processing. Sometimes textual QA is also called corpus-based QA.

Textual QA systems may be either *open* (also called general) *domain* systems or *closed* (also called restricted) *domain* systems. Open domain systems take as input all kinds of questions. Closed domain QA systems restrict themselves to a specialized domain, such as the medical domain or a company's products. The experiments presented in this thesis deal with open domain question answering, but the methods presented could as well be used in a closed domain system.

In practice, rare systems are purely textual, as it makes sense to store already extracted answers to *frequently asked questions* and to compare new questions for similarity with the old ones as well as to use structured data in parallel with unstructured data if available. There are QA systems that are solely based on comparing the new incoming question to previously asked questions [BHK+97, AHK+03, TL04]. These systems direct all new questions – that is all questions that are detected to be very dissimilar form the previous ones – to a human who provides the answer. This type of systems are especially good for cases where questions with the same semantic contents tend to be asked often. When using the documents from the WWW as a database for finding answers, the structured parts in the documents are also very useful and not only the unstructured text. Answering questions from structured or semi-structured data naturally requires different techniques than answering questions based only on unstructured text [LK03]. QA systems that are based on *structured or semi-structured data* are based on traditional work on natural language interfaces to relational databases.

Another dimension of QA systems are *cross-language systems*. They may either be systems that take the input question in one language, translate it or only the relevant query terms into a target language. The text database is in the target language and the answer to the question is also returned in this language. This is the type of systems that are evaluated at the CLEF QA evaluation forum, and it presupposes that the users of the system have a good passive knowledge of the target language. The approach taken at the NTCIR QA systems evaluation campaign is similar to that taken at CLEF except that the answers are translated back to the source language.

Question reformulations have been widely used in textual QA to produce answer extraction patterns. For example, in the work of Yousefi and Kosseim [YK06], question reformulations are used to produce patterns that extract the answer from semantically similar text snippets. Their QA system does not have a question classifier. Instead, an incoming question is mapped to a set of question reformulations that correspond to a set of answer extraction patterns formed from them. Hovy et al. [HHR02] provide

a fine-grained question/answer typology with answer extraction patterns. A QA system using this typology and patterns is based on recognizing the different reformulations of the same question.

The questions that a question answering system answers can be categorized in many ways. In this thesis, the questions are categorized into factoid, definition and list questions. *Factoid* questions are questions whose answers typically are short facts. They often consist of only a few words. Factoid questions may be classified into several subcategories and into NIL questions. In this thesis, the subcategories given in the Multinine Corpus are used. They are: *Location*, *Measure*, *Organization*, *Other*, *Person* and *Time* [VMG+06]. The names of the classes are self-explanatory except perhaps for the class *Other*. The class *Other* comprises all factoid questions that do not fall into any of the other categories. Questions belonging to all categories and subcategories may also be NIL questions. A *NIL question* is a question that does not have an answer in the given document collection. A *Definition question* is a question that asks for a definition. Answers to definition questions are typically longer than answers to factoid questions. However, in this thesis, definition questions are not longer than one sentence. In this thesis, definition questions may be subcategorized only into the subclasses *Organization* and *Person*. Definition questions also may be NIL questions. *List* questions are questions whose answers are lists. Each answer of the list typically resembles an answer to a factoid question. A list question may be subcategorized in the same way as factoid questions and it may also be a NIL question. An answer to a list question may be assembled from different parts of a document collection.

2.1.2 A general system architecture

Textual QA systems may have several different types of architecture. They often consist of a core part that is common to almost all textual QA systems and several optional parts. The core part has stayed the same for several years and it is typically composed of a question processor, a document retriever and an answer processor [VD05, HM03]. The question processor typically performs question classification according to answer type and formulates the query for the document retriever based on an analysis of the question. The document retriever typically executes the query and retrieves either entire documents or passages. The answer processor usually extracts answer candidates from the retrieved documents or passages and selects the answer to be returned by the system. The system implemented as a part of this thesis complies to this core architecture, and it is described in more detail in Section 4.1. In addition to this core part described above, many QA

systems contain optional parts that enhance their performance. Typical examples of these optional parts are the processing of common question types offline, the identification of question reformulations, exploitation of the vast amount of text documents available in the WWW and the integration of a translation component for performing cross-language QA.

When questions are processed offline, the task resembles that of information extraction (IE), where a table of a database is filled with information found in text documents [GW98]. Processing frequently occurring questions offline and placing the answers into a relational database presents at least two advantages when compared to only using the core QA system components: firstly, the on-line processing time is cut down and secondly, methods already tested and proven to yield good results in IE can be directly applied to QA. This approach takes into account the need of processing frequently occurring question types in a different way from more rarely occurring question types. This approach has been employed at least by Fleischman et al. [FHE03] and Jijkoun et al. [JdRM04].

The second type of addition to a core QA system architecture is the exploitation of question reformulations. Question reformulations can be used in at least two different ways: if the system is built on data consisting of pairs of questions and answers, it makes sense to detect similarities between the new incoming question and the already existing questions in the database. If the new question is found sufficiently similar with an existing one, the same answer as for the existing one may be given to the new one. Another use of question reformulations is to process all questions into a canonical form to ease further processing, i.e. question classification and query formulation. Question reformulations have been studied from this second perspective at least in Aunimo and Kuuskoski [AK05]. From the above mentioned uses of question reformulations the first one is very common in FAQ-type systems and in systems built for company helpdesks [AHK+03, BSA00, TL04]. The major challenge in building this type of systems is to develop methods for recognizing and measuring similarity among different questions[HMP+01].

A third and very popular extension to the core QA system architecture is the exploitation of the vast amount of text available in the WWW. Especially systems participating in the TREC 2005 QA system evaluation campaign made significant use of the Web [VD05]. Some systems would search for the answer from the WWW and then append the answer to the query terms in order to search for a supporting document from the document collection used at TREC. Other systems would first search the answer from the document collection and then use the evidence found in

the documents of the WWW to rank the answers.

The fourth extension to the core QA system architecture is the integration of a translation component for performing cross-language QA. The translation component may be added before the question processor, in which case machine translation techniques are used to translate the whole question. Alternatively, the translation component may be added inside the question processor component in which case typically only query words are translated. If the answers retrieved by the system are translated back to the language of the question, another translation component is added into the answer processor.

However, even if the above mentioned additional methods are often exploited in textual QA systems, the main methods applied are still methods for extracting the answer to a question from a large amount of unstructured text data. These methods are often based on or somehow similar to methods used in information retrieval, information extraction and text mining.

2.2 Evaluation

The evaluation of QA systems means assessing the performance of different systems and ranking them accordingly. This demands for common benchmarking data and common measures of performance. These issues have been addressed by several evaluation campaigns for QA. The major campaigns only assess systems based on unstructured text. The approaches presented in the systems participating in these campaigns typically rely on both IR methods and natural language processing methods. In Europe and Asia, special emphasis has been put on developing and evaluating systems that perform QA in several different languages and on systems that perform even cross-language QA. There has been a recent effort for building an evaluation campaign for QA systems based on structured XML data [WG04, GW06].

The following subsections introduce evaluation campaigns and evaluation measures for QA systems based on unstructured data. Limiting ourselves to only this type of QA systems is justifiable because the new answer extraction methods presented in this thesis are designed for this type of systems and they are also evaluated using data and measures commonly used for QA systems based on unstructured data.

2.2.1 Evaluation campaigns

Evaluation campaigns are typically organized and funded by public organizations such as NIST (National Institute of Standards and Technology, Universities, JSPS (Japan Society for Promotion of Science) and ELDA (Evaluation and Language Resources Agency). The campaigns are open to any organization provided that it agrees not to use the data for other purposes than the evaluation of the QA system. All participants are strongly encouraged to describe in a publication the inner workings of their systems and the details of the methods it uses. In the following, four QA evaluation campaigns will be outlined: the TREC, CLEF, EQueR and NTCIR QA challenges.

The first evaluation campaign for QA systems was organized by NIST as a track in TREC [1] (Text REtrieval Conference) in 1999 [VD05]. The QA track has been going on since then. The tasks have evolved and the number of participants has grown steadily. The language in the QA track has been English. The TREC 2005 QA track contains three tasks: the main task, the document ranking task and the relationship task. The main task – also known as the information nuggets task – consists of a question series seeking for specific bits of information concerning a set of targets. The targets can be of four types: *Event*, *Person*, *Organization* or *Thing*. Each series consists of *Factoid* and *List* questions. The last question in every series is an *Other* question, which is defined as a question that asks for additional information not covered by the preceding questions. An example of a main task question series is given in Figure 2.1. We can observe from the example that the ordering of the questions is important not only for the last *Other* type of question, but also for questions containing anaphoric references to a previous question. An example of an anaphoric reference to the previous question is in the second question of Figure 2.1: *Where is his tomb?* The anaphoric pronoun *his* is a reference to the noun phrase *Imam of the Shiite sect of Islam* of the previous question.

The second type of task in TREC 2005 is the document ranking task. It uses a set of questions that is a subset of the questions from the main task. The goal of the participating systems is, for each question, to produce a ranked list of documents containing the answer. The third type of task in TREC 2005 is the relationship task. In this task, systems are given statements (called topics) for which they should provide evidence. Figure 2.2 provides an example topic of the relationship task along with example evidence that the systems may provide. Each evidence is marked by the assessors as vital or okay, according to how relevant it is as a piece

[1] http://trec.nist.gov

of evidence for the topic.

	Target of type *Thing*: Shiite
Factoid	Who was the first Imam of the Shiite sect of Islam?
Factoid	Where is his tomb?
Factoid	What was this person's relationship to the Prophet Mohammad?
Factoid	Who was the third Imam of Shiite Muslims?
Factoid	When did he die?
Factoid	What portion of Muslims are Shiite?
List	What Shiite leaders were killed in Pakistan?
Other	

Figure 2.1: Example of a TREC 2005 main task question series. The target of all of the 8 questions is Shiite. In front of each question is its type. The query of type *Other* is at the end of every question series and it does not have any specific question string.

Topic: The analyst is concerned with arms trafficking to Colombian insurgents. Specifically, the analyst would like to know of the different routes used for arms entering Colombia and the entities involved.

Vital?	*Nugget of Evidence*
Vital	Weapons are flown from Jordan to Peru and air dropped over southern Columbia
Okay	Jordan denied that it was involved in smuggling arms to Columbian guerrillas
Vital	Jordan contends that a Peruvian general purchased the rifles and arranged to have them shipped to Columbia via the Amazon River.
Okay	Peru claims there is no such general
Vital	FARC receives arms shipments from various points including Ecuador and the Pacific and Atlantic coasts
Okay	Entry of arms to Columbia comes from different borders, not only Peru

Figure 2.2: Example of a TREC 2005 relationship topic and nuggets of evidence provided as answers by the systems.

The evaluation campaign whose data is used to evaluate the new answer extraction methods introduced in this thesis is the CLEF (Cross Language Evaluation Forum) evaluation campaign. The first CLEF evaluation campaign was organized in 2000 and its main mission has been to support

European research on information retrieval and on European languages. It has had a QA track since 2003. The QA track of CLEF has always provided monolingual and cross-language QA tasks for several languages. However, even though English has been available as both a source and target language in the cross-language tasks, it has not been available as a monolingual task in order to keep the TREC and CLEF data and tasks clearly different.

Another European QA evaluation campaign effort was the EQueR evaluation campaign for monolingual French language QA systems. It lasted for over three years between 2002 and 2006. In 2006 EQueR was stopped and the French participating organizations moved their efforts to the CLEF QA Track, which has provided a monolingual task for French and several cross-language tasks involving French since 2004. EQueR provided two tasks: an open domain QA task and a closed domain one, which was the medical QA task [Aya05]. The question types were: *Definition*, *Fact*, *List* and *Yes / No*. The systems were asked to return either a short CLEF style answer or a text passage of at most 250 bytes.

The NTCIR Workshop which concentrates on north east Asian languages and enhances research in information access has provided a QA track since 2002 [FKM04a]. In 2001, the document collection was provided in Japanese and the questions (or topics) in Japanese and English. The QA task has been going on ever since and more tasks have been added. In 2006, the languages concerned were Chinese, Japanese and English [2]. Both monolingual and cross-language tasks were provided. Unlike CLEF, NTCIR also provides the monolingual English QA task. All answers in the NTCIR QA task have to be named entities, by which the organizers mean entities of the following types: *Organization*, *Person*, *Location*, *Artifact* (e.g. product name, book title, pact, law), *Date*, *Time*, *Money*, *Percent* or *Numex* (numerical expression other than date or percent). This list of entities is the one provided by the Information Retrieval and Extraction (IREX) project [SI99]. Before introducing the tasks in different languages, NTCIR provided three different subtasks: subtask 1 where the systems return either only one answer (QA challenge of the year 2002) or an answer of type *List* (QA challenge of the year 2003), subtask 2 where the systems return an ordered set of up to 5 answers and subtask 3, where the questions consist of plain questions and of a follow-up question which typically contains an anaphoric or elliptic expression [FKM03, FKM04b]. When new languages and the cross-language challenge were introduced in 2005, the subtasks were left out in order to otherwise simplify the task [SChCL05].

[2] For information on the NTCIR 2006 campaign, consult e.g. http://clqa.jpn.org/

Having the different subtasks made the NTCIR QA challenge resemble the TREC QA challenge, and introducing more languages and a cross-language task brought the NTCIR QA challenge closer to the CLEF QA challenge.

2.2.2 Measures for evaluation

The metrics that are used in the evaluation of the performance of QA systems are either the same metrics as those used in measuring the performance of information retrieval systems or they have been inspired by them. In general, these metrics measure only the quality of the answers returned by the system and not the time it takes for the system to produce the answer. The rest of this section will give the descriptions of the metrics that are used to measure the quality of the results.

The answers given by the QA system are categorized into three classes in the evaluation process: *right*, *wrong* or *inexact*. Right and wrong are self-explanatory, but inexact needs an explanation: It means those answers that are correct but either incomplete or contain superfluous words. For example, for the question *Who is Felipe Gonzales?*, an incomplete answer would be *prime minister* because the right answer is *the Spanish prime minister*. An example of an answer to the same question that contains superfluous words is *the Spanish prime minister and the Portuguese prime minister*.

In addition to binary relevance assessments where a right answer scores 1 and a wrong or inexact answer 0, an answer rank based score called *answer rank score*, is calculated. The answer rank score may also be called the *mean reciprocal rank* score. Answer rank based score calculation has been used in the TREC QA system evaluation [Voo99]. It assumes that a system returns a ranked list of answers. The performance score of the system with regard to a question is the reciprocal of the rank of the first appearance of a right answer in the answer list. In the case of a NIL question, the correct answer is either the string *NIL* or the empty space after the last item in the answer list. The answer rank score reflects how well the system can order the result set it returns, and in the case of NIL questions, it also reflects how well it can determine when to cut off the answer list. The first property requires that the question specific scoring works and the second property requires that the global scoring works, i.e. that the scores given for answers to different questions are comparable with each other. The scores given to answers returned by a QA system have also been called confidence scores, and there are also other measures that may be used to measure how well the scoring module performs. The most commonly used scores for this purpose is the *CWS*, confidence weighted score which is used

at the TREC and CLEF QA evaluation campaigns [Voo02, MV+05]. The correlation coefficient between the confidence value given by the system and the judgment of a human assessor have also been used, as well as the K1 score (see e.g. [MV+05, HPV05]).

The system's performance on all NIL-questions is reported as *precision (P)*, *recall (R)* (see e.g. [vR75]) and F_1*-measure*. Each individual NIL-question and NIL answer is simply judged as either *right* or *wrong*. Precision is defined as:

$$P = \frac{TP}{TP + FP}, \tag{2.1}$$

where TP is the number of true positives and FP is the number of false positives. In the case of NIL-questions TP is the number of NIL-questions correctly answered by the string *NIL*. The term FP means the number of non-NIL questions answered by the string *NIL*. The equation for recall is given in Equation 2.2.

$$R = \frac{TP}{TP + FN}, \tag{2.2}$$

where FN is the number of false negatives, i.e. the number of NIL-questions that did not receive the string *NIL* as an answer. Because precision and recall are related to each other – as one increases, the other tends to decrease and vice versa – F-measure is used to combine the information present in them into one measure. F_β-measure is defined in Equation 2.3.

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{2.3}$$

where β is is the parameter used to balance P and R. When β is one, precision and recall are given equal weight. When β is greater than one, recall is favored, and when β is less than one, precision is favored. In this thesis, when we report the performance of the system on NIL-questions, precision and recall are given equal weight. This is in line with the common practice at the CLEF evaluation campaign and thus makes the results easily comparable with those obtained by other systems evaluated with the same data. On the other hand, if we look at the importance of precision and recall from the end user's point of view, we could argue that recall should be given more weight than precision. This is because answering NIL to all NIL-questions and some non-NIL questions might be more desirable from the user's point of view than returning wrong non-NIL answers to both NIL-questions and non-NIL questions.

In this thesis, we do not calculate precision and recall for non-NIL questions. There are two reasons for this. Firstly, it is not a common practice at the QA evaluation campaigns and secondly, as the calculation of precision and recall for individual non-NIL questions is not feasible with the evaluation data at hand, no average figures for all non-NIL questions can be provided. Calculating precision and recall for the whole set of non-NIL questions in the same manner as is done for the NIL-questions would not be informative. We would then measure only how well the system distinguishes NIL and non-NIL questions and not how correctly the system answers to non-NIL questions.

Calculating meaningful precision and recall figures for single non-NIL question would require that we assume that the QA system returns a list of answers instead of just one answer. This is what information retrieval systems commonly do. They return a list of documents for a single topic. Now, determining the precision of a non-NIL question would be straightforward. In fact, the percentage of right answers can also be called precision. However, calculating recall for a non-NIL question is in general not feasible with the evaluation data provided. This is because it would be very difficult to determine the set of all correct answers to a question appearing in the text corpus. There are two reasons for this. Firstly, the set of correct answers is difficult to determine because it should also contain wrong answers appearing in the text, as these are treated as right answers in the evaluation. For example, if it is stated in the text that the president of Finland is Sauli Niinistö and not Tarja Halonen, then Sauli Niinistö has to be treated as a right answer. The second reason that makes it difficult to determine the set of all correct answers is that many questions have an unlimited number of possible correct answers. In order to be able to measure the recall of a single question, we would need a set of evaluation data where all possible correct answers are marked instead of just a few.

Chapter 3

State of the art in answer extraction

This chapter describes the state of the art in the extraction of *factoid and definition* answers in open domain textual QA. The answers are short text snippets, typically named entities or numeric or temporal expressions. The main approach is the pattern matching based approach. These methods are explored in detail in the next section. In the second and last section of this chapter, other methods are briefly introduced. Comparison of the methods presented in this chapter and of the novel methods presented in this thesis is given in Section 8.3.

3.1 Pattern matching based methods

The answer extraction methods based on pattern matching form the most simple commonly used group of answer extraction methods. The most complex part of the methods lies in the way the patterns are created or generated, whereas the patterns themselves are simple. Answer extraction methods based on pattern matching follow the tradition of information extraction methods. In the rest of this section we will first describe the existing answer extraction patterns, secondly present ways for forming them and last discuss the performance of the different approaches.

3.1.1 Format of patterns

Answer extraction patterns (AEPs) may consist of several types of units, such as punctuation marks, capitalization patterns, plain words, lemmas,

part-of-speech (POS) tags, tags describing syntactic function, *named entities (NEs)* and temporal and numeric expressions. Some AEPs contain one or several words that are extracted from the question [RH02, XWL04]. In the following is an example of a question, *Q*, and of a text snippet, *S1*, containing the answer for which an answer extraction method based on simple pattern matching typically is sufficient. The example has been taken from the Multinine Corpus [MV+05].

Q: What is UNITA?
S1: UNITA (the National Union for the Independence of Angola).

Different types of AEPs are described in the following. We start from the most simple ones containing plain words and punctuation marks and proceed on to more complex ones that may contain NEs and syntactic functions.

Plain words, punctuation marks and a question word are used by the patterns of Ravichandran and Hovy, 2002 [RH02] and by one of the best performing QA systems at CLEF 2006 [JGTVDC+06]. For example, Table 3.1 lists example patterns for the question classes *Inventor* and *Discoverer*. The $< ANSWER >$ tag shows the place of the answer to be extracted and the $NAME$ tag shows the location where the question string that has been identified as a proper name by a NE recognizer has to be inserted. The figure in the leftmost column is the precision of the pattern. The precision of a pattern is used in the process of pattern generation as will be explained in Subsection 3.1.2.

	Inventor
1.0	<ANSWER> invents <NAME>
1.0	the <NAME> was invented by <ANSWER>
1.0	<ANSWER> invented the <NAME> in
	Discoverer
1.0	when <ANSWER> discovered <NAME>
1.0	<ANSWER>'s discovery of <NAME>
1.0	<ANSWER>, the discoverer of <NAME>

Table 3.1: Examples of AEPs containing plain words, punctuation and one question word.

Another type of patterns are the *surface patterns*. They are *lexico-syntactic patterns* that are especially useful for handling frequent question types such as Who is ..., Where is ..., What is the capital of ...,

When was ... born?. This type of patterns are widely and successfully used [JMdR03, SS01, SS02]. Two examples of lexico-syntactic patterns (*P1* and *P2*) and of text snippets (*S1* and *S2*) matching them are presented in the following:

Q: When was NAME born?
P1: NAME was born in $< a >$YEAR$< /a >$
S1: that Gandhi was born in 1869 on a day which ...
P2: NAME ($< a >$YEAR$< /a >$-YEAR)
S2: In India, Gandhi (1869-1948), was able to ...

The capitalized words in the question and the patterns mean NEs or expressions. The patterns work so that when a new question such as *When was Gandhi born?* comes into the system, it is first analyzed and the string recognized as a NE of type *NAME* (i.e. *Gandhi*) is inserted into the patterns. Then the patterns are matched against the text snippets. The words inside the $< a >$ tags show where the answer is found. In the examples above, the answer has to be recognized by the expression analyzer as being an expression of type $YEAR$.

A third type of patterns are the *POS patterns*. In the following is an example of two POS patterns, *P1* and *P2* for handling questions beginning by *Who is* along with example text snippets, *S1* and *S2* that match the patterns [FHE03]:

P1: NNP* VBG* JJ* NN+ NNP+
S1: ABC/NN spokesman/NN Tom/NNP Mackin/NNP
P2: NNP+ , DT* JJ* NN+ IN* NNP* NN* IN* DT* NNP* NN* IN* NN* NNP* ,
S2: George/NNP McPeck/NNP, an/DT engineer/NN from/IN Peru/NN,

The syntax of the above POS patterns is that of regular expressions, i.e. the symbol + means at least one and the symbol * means none or any number of occurrences. The POS abbreviations such as *NNP* and *VBG*, come from the POS tagger, which uses the tag set of the Penn Treebank [MSM93] from LDC. The meanings of the tags used in the example patterns above are as follows:

DT determiner
IN preposition or subordinating conjunction
JJ adjective

NN noun, singular or mass
NNP proper noun, singular
VBG verb, gerund or present participle

The fourth type of AEPs are called *syntactic patterns*. They make use of either parse trees or parse dependency graphs to determine whether the phrase identified by a NE, numeric and temporal expression recognizer occurs in the right position. For example, let us examine the following question and the following answer text snippets:

"Q: Who developed the vaccination against polio?
S1: Dr. Jonas Salk, who developed a polio vaccine ...
S2: Dr. Albert Sabin, developer of the oral polio vaccine, ... " From Monz [Mon03]

In the first answer phrase of the example above, the answer is expressed as the subject of a relative clause. In the second answer phrase, it is expressed as a noun phrase which is modified by an apposition. Such syntactic patterns have been used by Jijkoun et al. [JdRM04] and Katz and Lin [KL03], among others. Example syntactic patterns along with example text snippets that they match are shown in Table 3.2. The syntactic dependencies are shown as arrows from dependents to heads. The name of the dependency in question is shown in the upper right corner of each arrow. For example, *Joseph Beard* is the head and it is a NE of type *person* and *a major developer* is an apposition that is dependent of the head. The arrow says that the apposition that is a dependent of the head is the role. The patterns use a NE recognizer and lists of words extracted from the WordNet. These lists include lists for possible *roles*, such as *major developer* and lists for possible *role-verbs*, such as *inventor*.

Pattern	*Example text snippet*
Apposition person $\longrightarrow^{app}$ role	a major developer, Joseph Beard
Apposition person $\longleftarrow^{app}$ role	Jerry Lewis, a Republican congressman
Clause person $\longrightarrow^{subj}$ role-verb	Bell invented the telephone

Table 3.2: Examples of syntactic AEPs for extracting roles [JdRM04].

The fifth and last type of AEP that is introduced here is the most complex one. This type of AEPs are used by the best performing system in the CLEF evaluation campaign of the year 2006. The AEPs consist of POS tags, NEs, collocations, expressions, semantic entities and words of

an ontology [ALM+04, LSN06, MMM+06]. Table 3.3 shows two example patterns (P1 and P2) along with example questions (Q1 and Q2) related to the patterns. The first question and pattern belong to the class *Function*. The pattern contains one slot, <FUNC1>, that is filled by language-specific names of professions and functions that come from an ontology. The second token of the pattern may be a proper noun or a NE. Thus, the pattern would match a text snippets such as: *President Malouwid* and President Mr. Jack Tim Malouwid. The second question and pattern belong to the class *Birth date*. The pattern contains one slot, <BIRTH1> or <BIRTH2> that is filled from language-specific words from an ontology. The slot <BIRTH1> may be filled by words such as *birth* and the slot <BIRTH2> may be filled by collocations such as *is born* and *was born*. The first token of the pattern is an expression of type *date*. Thus, the pattern would match text snippets such as *10.12.1997 was born*, *10.12.1997 is born* and *10.12.1997, the birth*.

	Function
Q1:	Quem é o presidente da Albânia? (Who is the president of Albania?)
P1:	<FUNC1> + proper noun or NE <FUNC1> = profession, function etc.
	Birth date
Q2:	Quando é que nasceu a Dolly? (When was Dolly born?)
P2:	date + <BIRTH1> or <BIRTH2> <BIRTH1> = birth, etc. <BIRTH2> = is born, was born, etc.

Table 3.3: Examples of AEPs containing plain words.

All of the AEPs described above are based on a fine-grained question classification. This is also the case for many well-performing QA systems. For example, the best performing QA system for monolingual French QA at CLEF 2006 is based on a question classification that contains 86 different classes [LSN06]. Each of these classes has a different set of AEPs. Using such a fine-grained question classification means that the AEPs may be very specific.

3.1.2 Creation of patterns

The creation of AEPs is done either manually or automatically. For example, the QA system from Fleischman et al. [FHE03] uses only the two answer extraction patterns that are given in the previous subsection as an example of POS patterns. These patterns are naturally manually created. In their paper, they mention that the patterns are "quick and dirty", and that

they aim at high recall instead of high precision. The surface (also called lexico-syntactic) patterns used by Jijkoun et al. have also been created manually. In their experiments on Dutch [JMdR03], they report that they hand-crafted a small amount of patterns for extracting information about seven fixed categories such as currencies, leaders and roles. In addition, they also used existing knowledge bases such as the EuroWordNet[Vos98] to expand the patterns automatically. For example, in the pattern *NAME, clause involving a profession*, which is used to extract roles, the part *clause involving a profession* is replaced by the list of 900 names of professions found in the Dutch EuroWordNet. The syntactic patterns illustrated in Table 3.2 are also crafted manually along the same lines as the surface patterns.

The first paper that reports a method for generating answer extraction patterns automatically is that of Ravichandran and Hovy [RH02]. Their method is based on a careful classification of questions. For example, questions of type *Birthdate* form the class of questions that ask for somebody's birthdate. The data for forming the patterns is retrieved from the WWW and suffix trees (see e.g. [Gus97]) are used for forming the patterns from the sentences extracted. The suffix tree is used to find all substrings and their counts. After that, the precision of each pattern is calculated and patterns with a sufficiently high precision are retained.

One of the best performing QA systems at CLEF 2006 uses a sequence mining technique for pattern generation [JGTVDC$^+$06] for definition questions. The patterns are generated by first collecting relevant text snippets from the WWW and then using a frequent sequence mining technique to form the AEPs. The system uses a naïve Bayes classifier with carefully chosen features to perform answer extraction for factoid questions.

3.1.3 Performance of patterns

Comparing the performance of the different pattern types is not an easy task because they are seldom tested with the same data, and even if they are, QA systems typically are quite complex and there are also many other components besides the answer extraction component that affect the performance. However, below are some figures that the authors have reported on their methods.

The performance of the hand-crafted POS patterns from Fleischman et al. [FHE03] is 45% correct for pattern *P1* and 79% correct for pattern *P2*. However, these figures only report the correctness of extracted relations for questions of type *Who is ...* that might potentially be asked. This is basically an information extraction task where recall is not measured at all.

The data used in this experiment is a text database of 15 GB consisting of newspaper text from the TREC 9 and 2002 corpora, the Yahoo! news, the AP newswire, the Los Angeles Times, the New York Times, Reuters, the Wall Street Journal and from various on-line news websites. Out of all results extracted by the patterns, 5000 items were evaluated, and the results of this evaluation are reported above.

Using the surface patterns in their QA system, Jijkoun et al. managed to produce a correct answer to 22% of the CLEF 2003 [MRV+04] non-NIL questions. When testing the lexico-syntactic or surface patterns on the TREC 2002 and 2003 data and only on questions asking for roles, the percentage of correctly answered questions was 9% [JdRM04]. Using the same type of patterns, Soubbotin and Soubbotin managed to answer correctly 56.5% of the TREC 2002 [VH01] questions. Their system was the best one that year. Using the syntactic patterns and only questions about roles, on the data of TREC 2002 and 2003, the system managed to answer correctly 17% of the questions [JdRM04].

The mean reciprocal rank (see definition in Section 2.2.2, beginning on page 15) of Ravichandran and Hovy [RH02] is 0.382, which was obtained by selecting questions belonging to 6 different classes from the TREC 2001 corpus. This is far from the results of Soubbotin and Soubbotin, which was 56.5% of correct answers on all questions of the whole dataset.

3.2 Other methods

While the best performing systems at CLEF and NTCIR QA evaluation campaigns are based on pattern matching techniques and while these techniques along with methods for automatically creating the patterns are becoming more and more popular, the best performing system at TREC 2005 QA evaluation campaign is based on a set of different techniques [VD05, HMC+05]. It uses a semantic parser, a temporal context identifier and a logical prover. The logical prover has access to knowledge of five different types: extended WordNet axioms, ontological axioms, linguistic axioms, a semantic calculus and temporal reasoning axioms.

Below is an example of a question and of a text snippet containing the answer where complex reasoning on the lexical definition of a word is required. This is a case where simple pattern matching techniques would be prone to fail.

"Q: Where did Bill Gates go to college?

S1: Bill Gates, Harvard dropout and founder of Microsoft, ... " From Harabagiu et al [HMP+01]

The fact that Bill Gates has attended Harvard can be inferred from the noun *dropout*. However, drawing this inference automatically is quite complicated. First, the system could look for more information on the word *dropout* from a machine-readable dictionary such as the *WORDNET* [Fel98]. The entry for *dropout* in *WORDNET* states that it is *someone who quits school before graduation*. From this information, the system has to be able to make the inference that the verb *to quit* presupposes a prior phase of attending.

Besides using a logical prover to verify the answer, other methods for answer extraction in textual QA are also used. One type of methods are the very simple proximity based methods. They are often used as a fall-back strategy if none of the above mentioned methods succeeds. The principal idea of these methods is that the answer phrase has to occur in the same sentence as some of the query terms or in the preceding or following sentence.

Chapter 4

Proposed question answering system and data

In the two preceding chapters we have looked at the central concepts of textual QA systems, at ways of evaluating them and at the state-of-the-art in answer extraction. Now it is time to move on to examine the QA system into which the novel answer extraction methods have been incorporated and the data from which the AEPs are extracted and with which they are tested. This chapter also defines a measure for estimating the difficulty of a given question with regard to a specific data set. This measure is then used to assess the difficulties of the datasets used for training and testing.

4.1 System architecture

The architecture of the QA system that was developed in order to test and evaluate the new answer extraction methods is presented in Figure 4.1. The architecture is a typical architecture for a search engine based QA system that can handle monolingual QA tasks in several languages. Harabagiu and Moldovan present the architecture of a typical QA system [HM03]. In general, search engine based QA systems have three components: *Question Processing*, *Document Retrieval* (or *Document Processing* as Harabagiu and Moldovan call it) and *Answer Processing* (or *Answer Extraction and Formulation* as called by Harabagiu and Moldovan). These three components will be described in the following.

In our QA system, the *Question Processing* component consists of a *Language Identifier*, *Question Classifier* and *Question Normalizer*. The *Language Identifier* recognizes the language of the question – in our case English, Finnish or French – and passes this information on to all other

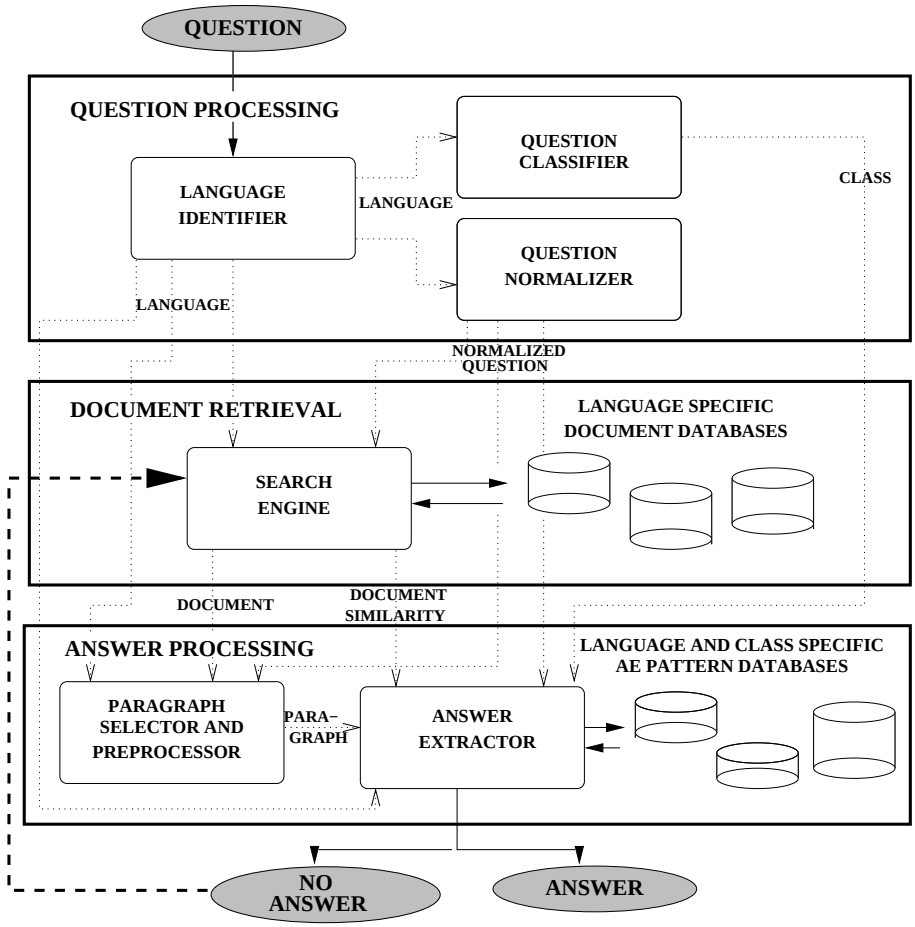


Figure 4.1: System architecture of the QA system.

modules of the system. In the figure, this is illustrated by the dotted arrows labeled with *LANGUAGE* that go from the box labeled *LANGUAGE IDENTIFIER* to all the other boxes. The task of the *Question Classifier* is to determine the expected answer type of the question. In our case the set of possible classes is: {*LOCATION*, *MEASURE*, *ORGANIZATION*, *ORGANIZATION DEFINITION*, *OTHER*, *PERSON*, *PERSON DEFINITION*, *TIME*}. This classification is taken from the Multinine Corpus [VMG$^+$06]. In our system, the class information is used only by the *Answer Extractor*. The *Question Normalizer* prepares the question into a format that can directly be used to form the query for the search engine, to select and preprocess the appropriate paragraphs from the documents returned by the search engine and to instantiate the answer extraction patterns. The *Ques-*

tion Normalizer prunes stop words (i.e. frequently occurring words that do not convey any content) from the questions and performs a simple recognition of multi word units. The stop word list used is the stop word list from SMART [Sal71].

The *Document Retrieval* module consists of an off-the-shelf search engine that retrieves documents from a language specific document database. Our system uses the open source search engine Lucene[1] [HG04]. The search engine may retrieve documents either from the English, Finnish or French document databases. Open source stemmers[2] are used both in indexing and in query term processing. For English, the Porter stemmer [Por80] is used, and for Finnish and French, Snowball stemmers [Por01] are used. The search engine is configured so that it does not exploit any stop word lists in search and indexing. The system requires that all query terms – a term may be a simple word or a phrase – appear in the documents retrieved. The search engine will be presented in more detail in Subsection 5.1. The formula for calculating the similarity score of a document with regard to the query is given in the same subsection in Equation 5.2. The similarity score is compared with a threshold value in order to determine which documents are passed on to the *Paragraph Selector and Preprocessor*. In addition, if the QA system finds no answer at all, the similarity score threshold value may be lowered and the execution of the system goes back to the *Document Retrieval* component. This is illustrated in Figure 4.1 by the dashed arrow pointing from the ellipse marked as *NO ANSWER* to the box representing the search engine. A dashed arrow is used instead of a normal arrow to illustrate that the execution of the system does not necessarily go back to the *Document Retrieval* component, but it may also stop there and return *NIL* as an answer. Which choice is made depends on the similarity score. The similarity score is also used by the *Answer Extractor* module to rank answers. The *Search Engine* module passes it there as is illustrated by the arrow labeled with *DOCUMENT SIMILARITY*.

The *Answer Processing* module consists of the *Paragraph Selector and Preprocessor*, of the *Answer Extractor* and of the language and class specific AEP databases. The main subject of this thesis is the method for the automatic induction of the AEPs from data and their application in our QA system. The *Paragraph Selector and Preprocessor* takes as input the documents retrieved by the *Search Engine* and selects those paragraphs that contain at least one query word. A paragraph may consist either of the title of a newspaper article or of a text paragraph as normally under-

[1] Java Lucene is available at http://lucene.apache.org/java/docs/index.html
[2] The stemmers are available at: http://www.snowball.tartarus.org/.

stood by the term. The selected paragraphs are then preprocessed into a compatible format with the AEPs. The *Answer Extractor* then takes each preprocessed paragraph and its associated *document similarity*, matches the relevant patterns to the paragraphs and scores the possible matches, or answer candidates. The system uses a document similarity value threshold to determine the documents to be processed. If the similarity score threshold value is reached and no answer is found, the system returns the string *NIL*, which indicates that it believes that the document collection contains no answer to the question. The document similarity threshold value was determined using two kinds of information in the training data. Firstly, the document similarity values of the documents containing right answers in the training data were analyzed, and secondly, the document similarity values of the documents that corresponded to queries formed from NIL questions were analyzed. As a result of this analysis, the document similarity threshold value was set to 0.4. Another case where the QA system returns *NIL* as an answer is when none of the appropriate AEPs match the text paragraphs retrieved by the *Search Engine*.

4.2 Description of the data

The data used in the development and testing of the answer extraction methods comes from three different sources: the Multinine Corpus [VMG+06], the Multieight-04 Corpus[3] [MV+05] and the CLEF multilingual comparable corpus [Pet06]. The first two corpora contain questions, answers and references to documents containing the answers in several languages. The third corpus contains newspaper articles in several languages. From all of these corpora, we only used the English language data. From the Multinine and Multieight-04 Corpora we chose those English questions that had an English answer. Also *NIL* was considered an answer. From these questions we pruned away those where the answer string contained a content word from the question. Two examples of such question answer pairs are presented in the following:

- PERSON D Who is Michel Noir? former Trade Minister *Michel Noir*, mayor of France's second city Lyon
- PERSON F Whose government broke off negotiations with the Tamil rebels, following Dissanayake's murder? Kumaratunga's *government*

[3]The Multinine and Multieight-04 Corpora can be downloaded from http://clef-qa.itc.it/downloads.html

In addition to pruning away those questions whose answers contain the answer string, also the class of some questions was changed. The changes made will be explained later on in this section. The complete data sets extracted from the Multinine and Multieight-04 corpora are listed in Appendix 1 and 2. The data in the corpora does not contain the named entity annotation that is present in the answers of the appendices. The English part of the CLEF multilingual comparable corpus consists of two document collections, the Los Angeles Times 1994 collection (425 MB, 113005 documents) and the Glasgow Herald 1995 collection (154 MB, 56472 documents) [Pet06]. This part of the corpus is called the Newspaper Corpus in this work.

As training data for inducing the answer extraction patterns, both the Multinine Corpus and the Newspaper Corpus are used. The Multinine Corpus contains questions and their answers, and the Newspaper Corpus contains answers and their contexts. As test data both the Multieight-04 Corpus and the Newspaper Corpus are used. The Multieight-04 Corpus contains a disjoint set of questions and answers from those of the Multinine Corpus. The answers to the test questions are extracted from the Newspaper Corpus.

The questions of the Multinine corpus are classified into 8 classes based on the expected answer type of the question. These classes and their abbreviations as used in this thesis are: *location* (*LOC*), *measure* (*MEA*), *organization* (*ORG*), *organization definition* (*ORGANIZATION D*, *ORGD*), *other* (*OTH*), *person* (*PER*), *person definition* (*PERSON D*, *PERD*) and *time* (*TIM*) [VMG+06]. The questions of the Multieight-04 are classified according to a question typology that contains two additional classes: *manner* and *object* [MV+05]. As the class *other* of the Multinine corpus comprises all other classes except the ones explicitly listed, the classes *manner* and *object* of the Multieight-04 are treated as if they were in the class *other*. This can also be seen from Appendix 2 which lists for each question its class, the actual question string and one right answer to the question. This data is used as the test data.

The question classification of the Multinine Corpus, which is based on the expected answer type, could suggest that all other answers except those belonging to the classes *ORGD*, *PERD* and *OTH* would be simple NEs, number expressions and temporal expressions as defined for example in the MUC-7 Named Entity Task [CR97]. These entities and expressions as well as their types are listed in Table 4.1. The entities and expressions of the MUC-7 NE Task are chosen for reference when analyzing the answers of the data because MUC-7 was the last MUC conference [Chi98], because its

Named entity	
Location	<ENAMEX TYPE="LOCATION">Berlin<ENAMEX>
Person	<ENAMEX TYPE="PERSON">Ann<ENAMEX>
Organization	<ENAMEX TYPE="ORGANIZATION">HY<ENAMEX>
Number expression	
Money	<NUMEX TYPE="MONEY">$2<NUMEX>
Percent	<NUMEX TYPE="PERCENT">100%<NUMEX>
Temporal expression	
Date	<TIMEX TYPE="DATE">2007<TIMEX>
Time	<TIMEX TYPE="TIME">10 a.m.<TIMEX>

Table 4.1: The MUC-7 classification and notation.

entities and expressions are more similar with the extracted answers than those introduced by its successor, the EDT (Entity Detection and Tracking) Task provided by ACE (Automated Content Extraction) Program [Lin05] and because the MUC-7 entities are widely known in the research community as there is a considerable amount of existing research and software for the MUC-7 type NE Task as it has been going on in one form or another since the early 1990's.

The difficulty of the task of extracting the exactly right answer snippet is illustrated by the fact that the correspondence between the MUC-7 entities and expressions and the question classes is quite low. One would expect that the question class *LOC* would completely correspond to the MUC-7 class *location*, that the question class *PER* would completely correspond to the MUC-7 class *person*, and so on. In fact, one would expect a complete correspondence for all other question classes except the classes *ORGD*, *OTH* and *PERD*. If this was the case, the proportion of MUC-7 entities and expressions in the training data would be 62,5% and in the test data 67,7%. However, in the real datasets, their proportions are only 48.7% and 60.4%, respectively. This can be observed from Tables 4.2 and 4.3. The abbreviations of the title fields correspond to the MUC-7 categories presented in Table 4.1 as follows: location (LOC), organization (ORG), person (PER), money (MON) and percent (%). As the data does not contain any TIMEX answers of type TIME, that category is left out of the table. The abbreviations used for question classes are the ones used throughout this thesis.

The mapping between Multinine classes and the MUC-7 entities and expressions is not straightforward as will be illustrated in the following. The reader may see the details of the mapping by consulting Appendices 1 and 2 where the answers have been annotated according to the MUC-7 guidelines. The question classes *LOC*, *PER* and *ORG* are often MUC-7 style NEs, the question class *MEA* may be a MUC-7 style numeral expression and the question class *TIM* a MUC-7 style temporal expression. One would expect

that the answers to questions belonging to the question classes *ORGD*, *PERD* and *OTH* would generally be something else than MUC-7 style NEs, number expressions or temporal expressions. However, answers to questions of type *ORGD* are often NEs, as can be observed from Tables 4.2 and 4.3. The following is an example of such a question and answer pair:

Example 4.1 *What is UNITA? the < ENAMEX TYPE = "ORGANIZATION" > National Union for the Total Independence of Angola </ENAMEX>*

Questions of type *MEA* often are not MUC-7 style number expression as one would expect. One reason for this is that MUC-7 style number expressions only include expressions of type money and percent. The following is an example of a question and answer pair that belongs to the question class *MEA* but where the answer string is not a MUC-7 style number expression:

Example 4.2 *How old is Jacques Chirac? 62.*

Another reason for the fact that the mapping between the question classes and the MUC-7 entities and expressions is not complete is that answers to questions may consist of more than one NE. This is illustrated by the following question answer pair of the question class *PER*:

Example 4.3 *Who were the two signatories to the peace treaty between Jordan and Israel? <ENAMEX TYPE="PERSON">Hussein</ENAMEX> and <ENAMEX TYPE="PERSON">Rabin</ENAMEX>.*

The requirement that the answer is a continuous text snippet extracted from the text also results in the fact that not all answers are clear cut MUC-7 style NEs that correspond to the question class. This is illustrated in Example 4.4, where the answer to the question of type *PER* is not a MUC-7 style NE of type person.

Example 4.4 *Which two scientists discovered "G proteins"? <ENAMEX TYPE ="PERSON"> Alfred G. Gilman </ENAMEX>, 53, of the < ENAMEX TYPE = "ORGANIZATION" > University of Texas Southwestern Medical Center </ENAMEX> in < ENAMEX TYPE = "LOCATION" > Dallas < /ENAMEX > and < ENAMEX TYPE = "PERSON" > Martin Rodbell </ ENAMEX >*

All the above examples are taken from the Appendices 1 and 2, which contain the training and test data. The answers in the data are annotated

according to the MUC-7 guidelines. Tables 4.2 and 4.3 show in figures how well the training and test data question classification may be mapped to MUC-7 entities according to the principles given above. The tables tell the number of answers that represent certain MUC-7 style NEs (i.e. *ENAMEX*), number expressions *i.e. NUMEX* or temporal expressions *i.e. TIMEX*. For example, Table 4.2 shows that in the question class *LOC*, there are 18 answers that are MUC-7 style NEs of type location. The last column of the table,*OTH*, designates those answers that do not fall into any of the MUC-7 categories. They are typically conjunctions of NEs (see example 4.3), phrases containing expressions belonging to several MUC-7 categories (see example 4.4) or phrases that belong only partially to a MUC-7 category or phrases that do not belong to any MUC-7 category at all (see examples 4.2). Articles and prepositions are not taken into account when the classification of answers to the MUC-7 categories. For example, the answer in the example 4.1 is considered as being of the MUC-7 category NE, type organization. In tables 4.2 and 4.3, the figure in square brackets tells the total number of question answer pairs in the corresponding question class. The figures given in parenthesis tell how many times the NEs or expressions occur in answers that do not fall into any MUC-7 style category. In the training data, the highest numbers of such occurrences are in the MUC-7 style categories location and organization – 16 occurrences in both categories. In the test data, the highest number of NEs or expressions that occur in answers not belonging to any MUC-7 style category as a whole are in the MUC-7 style category location (8 occurrences). An example of an answer that contains a MUC-7 style NE of the type organization, but that cannot be categorized as a whole under any of the MUC-7 style categories is presented in Example 4.5 . According to the question classification of the Multinine corpus, the answer belongs to the class *PERD*.

Example 4.5 *Who is João Havelange? < ENAMEX TYPE = "ORGANIZATION" > FIFA </ENAMEX > 's Brazilian president*

The above answer phrase is primarily classified under *OTH*. As it contains a MUC-7 style NE of type organization, it is classified also under that class, but in parenthesis following the notation of the tables.

In addition to changing all questions of classes *OBJECT* and *MANNER* in the Multieight-04 corpus into the class other, all question classes of both the Multieight-04 and the Multinine corpora have been revised. If the expected answer to a question corresponded to a MUC-7 entity or expression, the class of the question has been changed accordingly. In the following are two examples of such changes:

Question class	MUC-7 Style Classification						
	ENAMEX			NUMEX		TIMEX	OTH
	LOC	ORG	PER	MON	%	DATE	
LOC [21]	18 (6)	-	-	-	-	-	3
MEA [20]	(1)	-	-	2	5	-	13
ORG [14]	(1)	12 (1)	-	-	-	-	2
ORGD [21]	(3)	8 (4)	-	-	-	-	13
OTH [15]	-	-	-	-	-	-	15
PER [29]	(1)	(1)	18 (9)	-	-	-	11
PERD [24]	(4)	(10)	(1)	-	-	-	24
TIM [16]	-	-	-	-	-	15	1
ALL [160]	18 (16)	20 (16)	18 (10)	2	5	15	82

Table 4.2: The question class specific and overall numbers of answers in the **training data** (only non-NIL answers are considered) that represent a certain NE, number expression or temporal expression according to the MUC-7 categories.

Question class	MUC-7 Style Classification						
	ENAMEX			NUMEX		TIMEX	OTH
	LOC	ORG	PER	MON	%	DATE	
LOC [23]	23 (0)	-	-	-	-	-	-
MEA [17]	-	-	-	1 (0)	2 (0)	-	14
ORG [21]	-	19 (1)	-	-	-	-	2
ORGD [12]	(2)	5 (0)	-	-	-	-	7
OTH [32]	(4)	(2)	-	-	-	-	32
PER [24]	-	-	24 (0)	-	-	-	-
PERD [9]	(1)	(2)	-	-	-	-	9
TIM [26]	(1)	-	-	-	-	25 (1)	1
ALL [164]	23(8)	24 (5)	24	1	2	25(1)	65

Table 4.3: The question class specific and overall numbers of answers in the **test data** (only non-NIL answers are considered) that represent a certain NE, number expression or temporal expression according to the MUC-7 categories.

- OBJECT $\longrightarrow$ LOCATION What is the world's highest mountain? Everest
- OTHER $\longrightarrow$ ORGANIZATION What band contributed to the soundtrack of the film "Zabriskie Point"? Pink Floyd

4.3 Difficulty of the question answering task

In this work, the difficulty of a question is determined with regard to a given text document collection, and not based on the properties of the question *per se*. A question may or may not have an answer in a given document collection, and if it has one, the sentence in which it appears may or may not match closely the question – as is illustrated by the following example:

"Q: Who discovered America?
S1: Columbus discovered America.
S2: Columbus Day celebrates the Italian navigator who first landed in the New World on Oct. 12, 1492."[4]

If a question does not have an answer in the given document collection, it is called a *NIL question* [Voo99]. Otherwise it is called a *factoid* or a *definition question*, which means that the answer to the question is a relatively short fact or a definition.

The questions of a QA system can be divided into two groups: the group of NIL questions and the group of factoid and definition questions. When measuring question difficulty, questions belonging to these two groups are treated separately. A question belonging to the group of factoid and definition questions is considered easy if it matches closely the text snippet where the answer is. In fact, the main problem to be solved in QA is how to define a similarity metric that measures the similarity between a question and the sentence in which the answer is [EM03]. On the other hand, a *NIL* question is regarded easy if it does not correspond in any way to the sentences in the document collection.

The difficulty D with regard to a document collection c of a question q belonging to the group of factoid and definition questions is defined as the conditional probability that the answer terms occur if the question terms occur in the document collection. In order to form a difficulty value of 1 to maximally difficult questions and a value of 0 to minimally difficult questions, the conditional probability is subtracted from 1. More formally:

$$D(q,c) = 1 - P(X_{q.a_t}|X_{q_t}) = 1 - \left(\frac{|Xq.a_t \cap X_{q_t}|}{|X_{q_t}|}\right),$$

where $q.a_t$ is the set of terms of an answer a to question q, and q_t is the set of terms of q. The term $X_{q.a_t}$ stands for the set of documents in which the terms $q.a_t$ occur, and the term X_{q_t} stands for the set of documents in which the terms q_t occur. If the probability $P(X_{q.a_t}|X_{q_t})$ is 1, the question is easy,

[4] From Hermjakob et al. [HEM02]

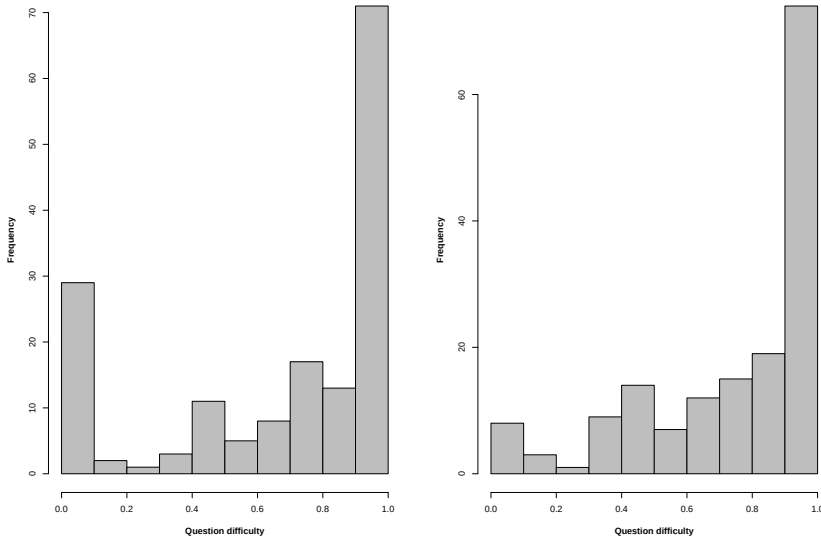
and if it is 0, the question is difficult. In order to calculate $P(X_{q.a_t}|X_{q_t})$, the number of documents containing both the answer terms and question terms $|X_{q.a_t} \cap X_{q_t}|$ and the number of documents containing only the question terms $|X_{q_t}|$ are calculated.

Figure 4.2 shows the frequencies of easy and difficult questions that belong to the group of factoid and definition questions in the training and test datasets. As can be seen from the figure, the highest columns for both datasets occur for questions with the difficulty value between 0.9 and 1. The average difficulty for the training data questions is 0.68 and for the test data questions it is 0.76. Table 4.4 shows for the same datasets the frequencies of questions that belong to the group of factoid and definition questions and that have the difficulty measures of 0 and 1. If a question belonging to the group of factoid and definition questions does not correspond to any document, its difficulty is 1 because it is impossible to extract the answer if no documents are provided. If a question belonging to the group of factoid and definition questions corresponds to only one document, and the answer is in that document, the difficulty measure is 0 and the question is regarded easy. Table 4.4 shows that the training dataset contains 24 and the test data set 13 factoid questions with no retrieved documents. On the other hand, the training dataset contains 15 easy questions belonging to the group of factoid and definition questions and the test dataset only 2 easy questions belonging to the same group. The variation in difficulty across classes inside the datasets is quite high, as can be seen from the box-and-whisker plots labeled *ALL* in the Figure 4.3. In addition, class specific question difficulty also varies in the training and test datasets. In the training data, the variation in question difficulty for the classes organization, other and person is especially high.

Number of docs		**0**		**1**
Question type		**NIL**	**F & D**	**F & D**
Difficulty		**0**	**1**	**0**
EN - EN training		11/20	24/160	15/160
EN - EN test		8/15	13/164	2/164

Table 4.4: Proportion of those *NIL* questions and of those questions belonging to the group of factoid and definition (F&D questions) questions that do not correspond to any documents as well as the proportion of *F&D* questions that correspond to exactly one document which contains the answer to the question.

The difficulty of a NIL question with regard to a document collection

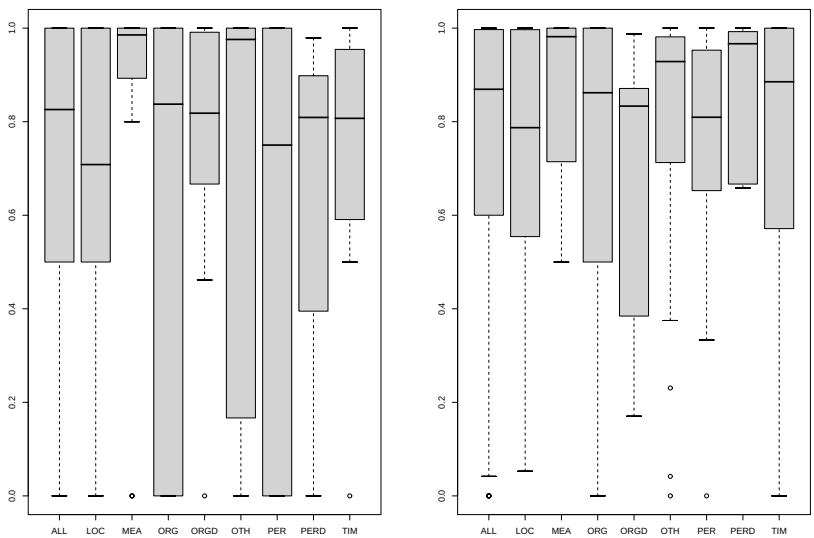


(a) English training data. Average question difficulty: 0.68.

(b) English test data. Average question difficulty: 0.76.

Figure 4.2: Histograms illustrating the frequencies of easy and difficult questions that belong to the group of factoid and definition questions for training and test data. The closer the difficulty measure is to 1, the more difficult the question is, and the closer it is to 0, the easier it is.

is determined by the number of documents the question words match. The more documents they match, the more difficult the question is. If the question words do not match any document in the collection, the question is trivially easy. The number of such trivial NIL questions in each dataset is shown in Table 4.4. The figure shows that in the training dataset 11 out of 20 and in the test dataset 8 out of 15 *NIL* questions are trivially easy.



(a) English training data. Median difficulty of all classes: 0.83

(b) English test data. Median difficulty of all classes: 0.87.

Figure 4.3: Box-and-whisker plots illustrating the class specific and overall distribution of questions into easy and difficult ones for training and test data. Only questions belonging to the group of factoid and definition questions are shown.

Chapter 5

Building blocks for the proposed methods

In the previous chapter we looked at the QA system into which the novel answer extraction methods will be incorporated and into the data with which they will be trained and tested. Before introducing the actual methods in Chapter 6, we will view the basic techniques on which the novel methods are built. These techniques include the vector space model used in document retrieval, agglomerative hierarchical clustering that is used for finding similar AEPs, edit distance that is used for measuring similarity and alignment that is used to generalize single AEPs.

5.1 Document retrieval in the vector space model

This section introduces document retrieval in the vector space model and explains how the *Document Retrieval* module (see Figure 4.1) of the QA system works. The *vector space model* is the most widely used document indexing and retrieval model. It was introduced by Gerard Salton in the early 1970s [Sal71]. In the vector space model, documents and queries are represented by term vectors. Document retrieval consists of calculating the similarity scores between the query vector and the document vectors and in returning a ranked list of the most similar document identifiers. The similarity of two vectors is defined as their dot product as presented in Equation 5.1.

$$similarity(\vec{q_k}, \vec{d_j}) = \sum_{i=1}^{N} t_{k,i} \times t_{j,i}, \tag{5.1}$$

where $\vec{q_k}$ is a query vector, $\vec{d_j}$ is a document vector, N is the number of

terms in the vector space, i.e. the number of distinct terms in the document collection, $t_{k,i}$ is the weight of the ith term of $\vec{q_k}$ and $t_{j,i}$ is the weight of the ith term of $\vec{d_j}$. There are several ways in which term weights may be determined. It is common to use binary term weights in query vectors and $tf \times idf$ weights in document vectors [Sal71]. The term *tf* means the *term frequency* of a term, i.e. the number of times the term appears in document d [Jon72]. The term *idf* is the *inverse document frequency* of a term, i.e. N/df, where N is the number of documents in the collection and *df* is the number of documents in which the term appears [Jon72]. To scale down the *idf* values, it is common to apply a logarithm function to it: $log(idf)$. The dot product between the query and document vector is typically normalized by dividing it by the lengths of the vectors, thus making similarity scores comparable across documents and queries with varying lengths.

We still have not addressed the issue of what is meant by terms even though we have discussed query and document terms. A simple and commonly used definition of a term is that it is a stemmed word or phrase that does not belong to the set of stop words. This is also the definition of a term that we have adopted in the *Document Retrieval* module of our QA system except for what concerns stop words. Our system does not prune away stop words in the indexing phase or in the document retrieval phase when the stop words are included in phrases. Here by phrases we mean all answer strings and all strings in questions that are surrounded by quotation marks. However, single terms that are stop words are pruned away from queries. For example, if the *Document Retrieval* system is given the following query:

```
+"of all the animals, man is the only one that is cruel" +who
+wrote,
```

it is transformed into:

```
+contents:"of all the anim man is the onli on that is cruel"
+contents:wrote.
```

As we can see in the example, stop words are not pruned away from the query phrase. The words *of, all, the, is, only, that* would be stop words according to a typical stop word list. However, the word *who* is pruned away as a stop word because it does not belong to a phrase. The query is formed from the natural language question *Who wrote: "of all the animals, man is the only one that is cruel"?* Using the entire phrase as a query term is important for the precision of the retrieval result.

The example query above also contains some other notations that need an explanation, namely the *+contents*. The + means simply that the term

has to appear in the documents retrieved. This is a way to raise the precision of the system. The *contents* is the name of the field from which the term is searched. Our system uses two fields, the *contents* field, which contains the body of a newspaper article, and the field containing the title of the article.

The similarity score produced by *Lucene*[1] [HG04], which is the *Search Engine* used by the QA system, follows the general principles presented in Equation 5.1 that are common when using the vector space model. The similarity between the query and the retrieved documents is calculated using the formula presented in Equation 5.2:

$$similarity(d,q) = \sum_{t \in q \cap d} tf(t) \times idf(t) \times lengthNorm(t.field\ in\ d), \quad (5.2)$$

where d is the document, q is the query and t is a term. The term $tf(t)$ is the square root of the *term frequency* of t, $idf(t)$ is the inverse document frequency of t and $lengthNorm(t.field\ in\ d)$ is $1/\sqrt{numTerms}$, where $numTerms$ is the number of terms in the field of document d where t appears. If the top-scoring document scores greater than 1, all document scores for the query are normalized from that score. This guarantees that all scores are real numbers between 0 and 1.

5.2 Agglomerative hierarchical clustering

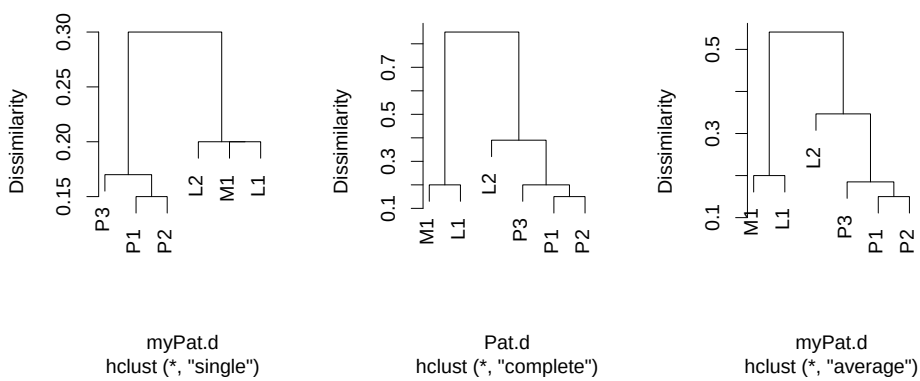
Agglomerative hierarchical clustering is a clustering method that, given a set of data samples and a matrix of dissimilarities (or distances) between them, in a bottom-up manner produces a grouping of the data that takes the form of a tree. Table 5.1 illustrates a dissimilarity matrix. The dissimilarity values range between 0 and 1, and the greater the value is, the more dissimilar two data samples are with each other. *Clustering* is a procedure that discovers subclasses of data samples that are more similar to each other than they are to other data samples. In the case of *hierarchical clustering*, the subclasses are represented as a tree where the leaves consist of all data samples, each forming a single cluster and where the root consists of one single cluster consisting of all the data samples. The rest of the tree consists of nodes containing from 2 to $n-1$ data samples that are most similar to each other, where n is the total number of data samples in the data set. These trees are often represented as dendrograms (see Figure 5.1)

[1] Java Lucene is available at http://lucene.apache.org/java/docs/index.html

or by a nested set of partitions (see Figure 5.2). *Agglomerative or bottom-up clustering* means that clustering begins by placing each data sample in its own cluster. At the following steps, the two most similar clusters are merged. This process of merging is continued until all data samples are in the same cluster. This amounts to $n - 1$ merges. Figure 5.3 illustrates the agglomerative clustering process. As can be seen in the figure, the clustering contains n levels or n different clusterings.

	M1	**L1**	**L2**	**P1**	**P2**	**P3**
M1	0	0.2	0.8	0.75	0.7	0.85
L1	0.2	0	0.2	0.3	0.33	0.4
L2	0.8	0.2	0	0.31	0.34	0.39
P1	0.75	0.3	0.31	0	0.15	0.2
P2	0.7	0.33	0.34	0.15	0	0.17
P3	0.83	0.4	0.39	0.2	0.17	0

Table 5.1: A dissimilarity matrix for the data samples MI, L1, L2, P1, P2 and P3.



(a) Single link clustering.

(b) Complete link clustering.

(c) Average link clustering.

Figure 5.1: Dendrograms illustrating three different clusterings of the data presented in Table 5.1

If at least one of the two clusters whose dissimilarity is to be calculated contains more than one data sample, there are several ways in which the

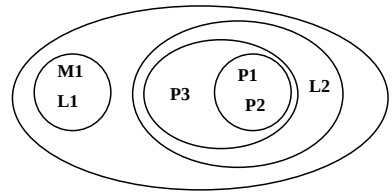


Figure 5.2: The clustering of Figure 5.1(c) represented as a Venn diagram.

Level	Average distance	Clusters
6	0.541	M1,L1,L2,P3,P1,P2
5	0.347	M1,L1 L2,P3,P1,P2
4	0.200	M1,L1 L2 P3,P1,P2
3	0.185	M1 L1 L2 P3,P1,P2
2	0.150	M1 L1 L2 P3 P1,P2
1	0	M1 L1 L2 P3 P1 P2

Figure 5.3: Agglomerative average hierarchical clustering. The clustering begins at the bottom where each data sample (M1, L1, L2, P1, P2 and P3) is in its own cluster and ends at the top where all data samples are in the same cluster.

dissimilarity between the clusters may be calculated. The most widely used ones among these are: *single*, *complete* and *average* link methods. They are presented in Equations 5.3, 5.4 and 5.5, respectively.

$$d_{single}(D_i, D_j) = \min_{x \in D_i,\ y \in D_j} d(x, y), \tag{5.3}$$

where D_i and D_j are clusters and $d(x, y)$ is the dissimilarity between the data samples x and y. The two clusters where the minimum similarity between two items is the smallest are merged.

$$d_{complete}(D_i, D_j) = \max_{x \in D_i,\ y \in D_j} d(x, y) \tag{5.4}$$

The two clusters where the maximum similarity between two items is the smallest are merged.

$$d_{average}(D_i, D_j) = \frac{1}{|D_i| \times |D_j|} \sum_{x \in D_i} \sum_{y \in D_j} d(x, y), \tag{5.5}$$

where $|D_i|$ and $|D_j|$ denote the number of data samples in the clusters D_i and D_j, respectively. The two clusters where the average similarity between all items is the smallest are merged.

Figure 5.1 illustrates the single, complete and average clusterings of the same data. In producing the dendrograms, a decision is needed at each merge to specify which subtree should go on the left and which on the right. Since, for n data samples there are $n-1$ merges, there are $2^{(n-1)}$ possible orderings for the leaves in the dendrogram. The algorithm used to produce the dendrograms orders each subtree so that the tighter cluster is on the left (the last, i.e., most recent, merge of the left subtree is at a lower value than the last merge of the right subtree) [BCW88]. Clusters consisting of single data samples are the tightest clusters possible, and merges involving two such clusters place them in the order in which they appear in the dissimilarity matrix given as input to the clustering algorithm.

The single link clustering merges the clusters $\{L2\}$ and $\{M1, L1\}$ because the minimum distance between the clusters is 0.2 and the other possible choices have greater minimum distances. This phenomenon is also called the chaining effect. The complete link clustering does not merge these clusters because the maximum distance between the clusters is 0.8 and because the maximum distance between the clusters $\{L2\}$ and $\{P3, P1, P2\}$ is smaller: 0.39. The average link clustering does not place *L2* into the same cluster with *M1* and *L1* because the average distance between the clusters is 0.5 while the average distance between the clusters $\{L2\}$ and $\{P3, P1, P2\}$ is 0.347. Sometimes hierarchical clustering is used to produce a single clustering and not a tree of n different clusterings. In that case, if the number of desired clusters is not known beforehand, the maximal gap in the dissimilarity values of the clustering may be used to determine which of the n possible clusterings produced is to be retained. Using this method to produce a clustering of the example data would yield the following results:

Single link Dissimilarity values at different levels: 0 0.15 0.17 0.20 0.20 0.30. Clustering at largest gap: $\{P3, P1, P2\}$ and $\{L2, M1, L1\}$.

Complete link Dissimilarity values at different levels: 0 0.15 0.20 0.20 0.39 0.85. Clustering at largest gap: $\{M1, L1\}$ and $\{L2, P2, P1, P2\}$.

Average link Dissimilarity values at different levels: 0 0.150 0.185 0.200 0.347 0.541. Clustering at largest gap: $\{M1, L1\}$ and $\{L2, P2, P1, P2\}$.

The single linkage method is good when the clusters are fairly well separated but they are not compact. The single linkage method is sensitive to the chaining effect, which occurs if there is a data point which is close to two separate clusters. The single linkage method joins these clusters. The complete linkage method is the opposite of the single linkage method. It performs well when the clusters are compact and roughly equal in size. The above methods are often used with some threshold value which means that clustering is stopped when the distance (be it minimum, maximum or average) between clusters exceeds a certain value.

In addition to agglomerative hierarchical clustering, top-down (also called divisive) hierarchical clustering methods are also widely used. They are more efficient than agglomerative methods when the desired number of clusters is relatively small because then the clustering may be stopped at an early stage, and vice versa, agglomerative methods are efficient when the desired number of clusters is relatively high. Another point in favor of using agglomerative instead of divisive clustering is that the computation needed to go from one level to the next in agglomerative clustering is simpler than in divisive clustering.

In most other widely used clustering methods, such as k-means (see e.g. [DHS01]), the EM algorithm [DLR77] and the Bayes classifier (see e.g. [DHS01]), the number of clusters to be formed has to be specified beforehand. Hierarchical clustering methods are useful for inspecting the data in order to determine the optimal number of clusters. The natural number of clusters may be obtained by inspecting the dissimilarity values of the cluster merges at the $n - 1$ levels of the clustering. The natural number of clusters occurs where there is an unusually large gap in the dissimilarity values. It there is no such gap, the method is not able to give any natural number of clusters, but all clusterings are more or less artificial. Hierarchical methods are also preferred when the clustering is dependent on a dissimilarity threshold and not on the number of clusters. These two benefits of hierarchical clustering may be obtained with the other clustering methods as well if they are applied separately to all possible numbers of clusters. However, in that case their efficiency drops drastically.

5.3 Edit distance and alignment

This section introduces the string method that is used in the AEP generation methods proposed in this thesis: the calculation of edit distance and the generation of an alignment. In general, string methods are commonly used when the data consists of ordered sequences of discrete symbols. These

ordered sequences are often also called strings, patterns or words, and the discrete symbols are often also called characters or letters. Besides edit distance calculation and alignment, other common string methods are different string matching methods. They are used to search for the exact or approximate matches of a string from a text, which is typically a particularly long string.

Before introducing edit distance [WF74] and alignment in detail, in Definition 5.1 we give the basic definitions for the concepts alphabet, string and character. The definition is followed by the introduction of the basic notations related to these concepts.

Definition 5.1 (alphabet, string, character) *An alphabet Σ is a finite ordered set. An element of Σ is called a character. A sequence of characters is called a string.*

Let us denote a string S by $s_1 s_2 \dots s_n$, where $s_i \in \Sigma$, and where the index i of s denotes the position of s_i in S. Let us denote the set of all strings over Σ by Σ^*. Thus, $S \in \Sigma^*$. The length of S is denoted by $|S| = n$ and the size of Σ is denoted by $|\Sigma|$. An empty character is denoted by ϵ and $\epsilon \notin \Sigma$.

Definition 5.2 (metric) *A metric $d(\cdot, \cdot)$ is a function that gives a scalar distance between its two arguments. A metric must satisfy the following axioms:*

1. *Nonnegative property: $d(a, b) \geq 0$;*
2. *Reflexivity: $d(a, b) = 0$ if and only if $a = b$;*
3. *Symmetry: $d(a, b) = d(b, a)$;*
4. *Triangle inequality: $d(a, b) + d(b, c) \geq d(a, c)$.*

Definition 5.3 (edit distance) *Given two strings $S \in \Sigma^*$ and $T \in \Sigma^*$, the edit distance $edit(S, T)$ is the smallest cost sequence of edit operations that is needed to transform the source string S into the target string T.*

Edit distance is also called Levenshtein distance[Lev66]. The basic edit operations for transforming S into T are:

Deletion: The character s_i does not correspond to any character in T, $s_i \rightarrow \epsilon$.

Insertion: The character t_j does not correspond to any character in S, $\epsilon \rightarrow t_j$.

Substitution: The character s_i corresponds to the character t_j and $s_i \neq t_j$, $s_i \rightarrow t_j$

Note that the substitution operations may not cross each other. This means that if $s_i \rightarrow t_j$ and $s_{i'} \rightarrow t_{j'}$, then $i < i'$ if and only if $j < j'$. Each operation has a cost. In the basic case, the cost of each operation is 1.

The edit distance measure has two parameters that may be altered: the costs of the operations and the operations themselves. The basic form of the edit distance measure is a metric, but some of its variations are not. Examples of variations of the edit distance measure that assign different costs to different operations are the insertion/deletion edit distance and the Editex phonetic distance measure. Examples of variations of the edit distance measure that differ in the set of operations are the Hamming distance and the Damerau-Levenshtein distance [Dam64].

In the insertion/deletion edit distance measure, the cost of the substitution operation is greater than 2 and the cost of the other operations is 1. This amounts to having just the two basic operations deletion and insertion, and to calculating the longest common subsequence (LCS) of the two strings. Sometimes edit operation costs are defined to reflect the nature of the data at hand. An example of this is the Editex phonetic distance measure [ZD96], which reflects the Soundex [HD80] and Phonix [Gad88] operations. More complex functions for assigning costs to edit operations have also been used. They have been based on phonetic feature tables or on assigning different costs to operations appearing at different locations in a string [ZD96].

In addition to assigning different costs to different edit operations, it is also possible to use the substitution operation only or to define new edit operations. An edit distance composed solely of the substitution operation is the Hamming distance. An example of a new operation is the *interchange* (also called transpose), which interchanges two neighboring characters. For example, the source *gabh* could be transformed into the target *gbah* with a single interchange edit operation. This distance is called the Damerau-Levensthein distance.

The sequence of edit operations needed to transform a source string into a target string can also be represented as a trace and as an alignment [Kru83]. These are illustrated in Figures 5.4 and 5.5, respectively. A trace from S to T consists of the source string S above and of the target string T below, usually with lines from some characters in the source to

some characters in the target. A character can have no more than one line, and the lines may not cross each other. Two characters are connected by a line only if they are the same or if they constitute a substitution operation. Source characters with no line represent a deletion and target characters with no line represent an insertion.

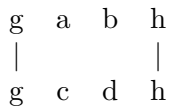


Figure 5.4: The trace from the source *gabh* to the target *gcdh*.

Definition 5.4 (alignment) *Given two strings $S \in \Sigma^*$ and $T \in \Sigma^*$, the alignment between S and T consists of a matrix of two rows. The upper row consists of the source S, possibly interspersed with null characters, denoted by ϵ. The lower row consists of the target string T, which may as well be interspersed with nulls. The column $\left[\begin{smallmatrix}\epsilon\\ \epsilon\end{smallmatrix}\right]$ of null characters is not permitted.*

Alignments are richer than traces because they make order distinctions between adjacent deletions and insertions. For example, the alignment given in Figure 5.5 is only one of the several possible alignments corresponding to the trace in Figure 5.4. The mapping from alignments to traces is many-to-one and onto, which means that several alignments may be mapped to one trace and that all alignments can be mapped to a trace.

The edit distance between a source and a target string can be calculated using the standard dynamic programming algorithm [WF74]. A distance matrix denoted by D is used by the algorithm. The algorithm produces the corresponding alignment or alignments as a side product. These are stored into the matrix $PRED$. The algorithm is listed in Algorithm 1. As input, the algorithm takes the strings S and T, whose lengths are m and n, respectively. As output, it returns the distance between S and T as well as the m-by-n matrix $PRED$ that contains all the possible alignments corresponding to the distance. Each cell in the $PRED$ matrix, except the cell PRED[0,0], contains at least one pointer to another cell of the matrix. The initialization phase consists of three steps, which are all listed on the

$$\begin{bmatrix} g & a & b & \epsilon & \epsilon & h \\ g & \epsilon & \epsilon & c & d & h \end{bmatrix}$$

Figure 5.5: One possible alignment between *gabh* and *gcdh*.

Algorithm 1 Edit distance with pointers.

Input: source string $S = s_1, s_2, ..., s_m$, target string $T = t_1, t_2, ...t_n$
Output: $D[m, n]$ and m-by-n matrix $PRED$

1: $D[0, 0] \Leftarrow 0$//Initialization begins.
2: $PRED[0, 0] \Leftarrow NULL$
3: **for** $i \Leftarrow 1; i \leq m; i++$ **do**
4: $D[i, 0] \Leftarrow D[i - 1, 0] + c(s_i \rightarrow \epsilon)$
5: $PRED[i, 0] \Leftarrow pointer\_to\_PRED[i - 1, 0]$
6: **end for**
7: **for** $j \Leftarrow 1; j \leq n; j++$ **do**
8: $D[0, j] \Leftarrow D[0, j - 1] + c(\epsilon \rightarrow s_j)$
9: $PRED[0, j] \Leftarrow pointer\_to\_PRED[0, j - 1]$
10: **end for**//Initialization ends.
11: **for** $i \Leftarrow 1; i \leq m; i++$ **do**
12: **for** $j \Leftarrow 1; j \leq n; j++$ **do**
13: $D[i, j] \Leftarrow min(D[i - 1, j] + c(s_i \rightarrow \epsilon), D[i, j - i] + c(\epsilon \rightarrow t_i), D[i - 1, j - 1] + c(s_i \rightarrow t_j))$
14: $PRED[i, j] \Leftarrow$ pointers to the cell(s) in $PRED$ that correspond to that/those in D based on which the value $D[i, j]$ can be calculated.
15: **end for**
16: **end for**
17: **return** D[m,n], PRED

lines 1 - 10 of the Algorithm 1. Firstly, the value of the cell $D[0,0]$ is set to 0, and the value of $PRED[0,0]$ is set to $NULL$. Secondly, the values of all the other first column cells are set to the value of the cell above incremented with the cost of the deletion operation on the corresponding character of S, and the corresponding cells in the $PRED$ matrix are assigned a pointer to the cell above. Thirdly, the value of all the other first row cells are set to the value of the cell on the left incremented with the cost of the insertion operation on the corresponding character of S and the corresponding cells in the $PRED$ matrix are assigned a pointer to the cell on the left. The values of the remaining cells are calculated using the values that have been calculated so far using the following recurrence:

$$D[i,j] = min \begin{cases} D[i-1,j] & + & c(s_i \rightarrow \epsilon) \\ D[i,j-1] & + & c(\epsilon \rightarrow t_j) \\ D[i-1,j-1] & + & c(s_i \rightarrow t_j) \end{cases}$$

where $c(s_i \rightarrow \epsilon)$, $c(\epsilon \rightarrow t_j)$ and $c(s_i \rightarrow t_j)$ are the character-specific costs for the deletion, insertion and substitution operations, respectively. This function is on line 13 of Algorithm 1. After having calculated a value of a cell in matrix D, the corresponding cell of the $PRED$ matrix is filled with a pointer to the cell from which this value was calculated. Sometimes there are several cells based on which the value of the cell in matrix D could have been calculated. In this case, pointers to all of them are stored into $PRED$. This is done on line 14 of the algorithm. After each cell in matrix D has been filled with a value, the edit distance can be found in cell D[m,n]. The alignment(s) corresponding to this value can be retrieved by traversing the $PRED$ matrix from the cell $PRED[m,n]$ to the cell PRED[0,0] . Figure 5.6 shows an example matrix that contains both the values of matrix D and the pointers of matrix $PRED$. In this matrix, the costs of the edit operations are defined as:

$$c(s_i \rightarrow \epsilon) = 1$$

$$c(\epsilon \rightarrow t_j) = 1$$

$$c(s_i \rightarrow t_j) = \begin{cases} 0 & if \quad s_i = t_j \\ \infty & otherwise \end{cases}$$

When the costs are defined as above, the algorithm calculates the insertion/deletion distance between the two strings. More complex cost functions for the operations exist for example in the Editex algorithm [ZD96]. In this algorithm, the costs for the operations also depend on the source and target character at hand.

		Target			
		g	c	d	h
Source	0	← 1	← 2	← 3	← 4
	↑	↖			
g	1	0	← 1	← 2	← 3
	↑	↑	↑	↑	↑
a	2	1	← 2	← 3	← 5
	↑	↑	↑	↑	
b	3	2	← 3	← 4	← 5
	↑	↑	↑	↑	↖
h	4	3	← 4	← 5	4

Figure 5.6: An example matrix containing both the integer values of the matrix D and the pointers to previous cells of the matrix $PRED$. This is produced while calculating the insertion deletion distance between the strings *gabh* and *gcdh*.

In the example matrix of Figure 5.6 we can observe that the insertion/deletion distance between the two strings is 4. As can be observed in the figure, there are several possible operation sequences that may be used to transform the source into the target and that have a cost of 4. The operation sequences can be represented as alignments, and they can be retrieved from the matrix by accessing cell D[m,n] and following all possible paths to cell D[0,0]. An arrow to the left represents an insert, an arrow to the northwest represents a substitution, and an arrow upwards represents a deletion. In the example matrix, a total of 6 different alignments can be found. These alignments are listed in Figure 5.7 Alignments have been used for example in grammar induction [vZ01] and in the generation of transformation rules for translating technical terms and proper names from one language to another [TPK$^+$05].

$$\begin{bmatrix} g & a & b & \epsilon & \epsilon & h \\ g & \epsilon & \epsilon & c & d & h \end{bmatrix} \quad \begin{bmatrix} g & a & \epsilon & b & \epsilon & h \\ g & \epsilon & c & \epsilon & d & h \end{bmatrix} \quad \begin{bmatrix} g & a & \epsilon & \epsilon & b & h \\ g & \epsilon & c & d & \epsilon & h \end{bmatrix}$$

$$\begin{bmatrix} g & \epsilon & a & b & \epsilon & h \\ g & c & \epsilon & \epsilon & d & h \end{bmatrix} \quad \begin{bmatrix} g & \epsilon & a & \epsilon & b & h \\ g & c & \epsilon & d & \epsilon & h \end{bmatrix} \quad \begin{bmatrix} g & \epsilon & \epsilon & a & b & h \\ g & c & d & \epsilon & \epsilon & h \end{bmatrix}$$

Figure 5.7: The six possible alignments between *gabh* and *gcdh* that have a cost of 4. The alignments may be retrieved from the matrix represented in Figure 5.6.

5.4 Multiple string alignment

This section provides the general background to the alignment based method developed in this thesis for the generation of AEPs from preprocessed text. Multiple string alignment is generally called multiple sequence alignment (MSA) and it is widely used in the analysis of biological data, such as protein sequences [DEKM98]. In this work, we only describe its usage on natural language data. In the previous section, we introduced the concept of an alignment between two strings. In the MSA terminology, the concept is called pairwise sequence alignment. The definition of a multiple alignment is a generalization of the definition for alignment, and it is given below:

Definition 5.5 (multiple alignment) *Given n strings $S_1, S_2, \ldots S_n \in \Sigma^*$, the alignment of the strings consists of a matrix of n rows. Each of the strings corresponds to one row. The strings may be interspersed with null characters, denoted by ϵ. A column consisting solely of null characters is not permitted.*

In the field of natural language processing, multiple alignment has been used in applications such as the creation of a lexicon of elementary semantic expressions using text datasets that supply several verbalizations of the same semantics [BL02] and the generation of sentence-level paraphrasing patterns using unannotated comparable corpora [BL03].

The most commonly used method to produce multiple alignments is *progressive alignment* [DEKM98]. It constructs a succession of pairwise alignments, and its main advantage is that it is fast and efficient. Several progressive alignment strategies exist, but most of them build a guide tree that is used to determine the order in which the strings are aligned. A *guide tree* is a binary tree whose leaves represent strings and whose interior nodes represent alignments. The root node represents a complete multiple alignment. The nodes furthest from the root represent the most similar pairs. The general algorithm for progressive multiple alignment is as follows:

1. Calculate a symmetric square matrix of $(n(n-1))/2$ distances between all pairs of n strings. The diagonal of the matrix consists of zeros.

2. Construct a guide tree from the diagonal matrix using an agglomerative hierarchical clustering algorithm.

3. Starting from the first node added to the tree, align the child nodes. The child nodes may consist of two strings, of one sequence and one

alignment or of two alignments. Repeat this for all nodes in the order that they were added to the tree until all sequences have been aligned.

The above algorithm does not define how the distances between pairs of strings is calculated, which agglomerative hierarchical clustering algorithm is used and how the alignments are performed. The specific algorithm that is used has to determine these. An example of a simple distance measure that can be used is the normalized edit distance between two strings. An agglomerative hierarchical clustering algorithm with different variants was described in Section 5.2. Alignments can be performed in many ways, as well. One way is to perform the alignment between two strings using the dynamic programming algorithm as described in the previous section. A new string can be added to an alignment (also called a group) by calculating its distance from each string in the group in turn. The nearest string belonging to the group is aligned with the new string. One way to align two groups is to calculate first all distances between the strings of both groups. Then the pair of strings with the smallest distance from each other and belonging to different groups is aligned. When aligning a string with a group or when aligning two groups, extra gap symbols might exist in the strings belonging to the groups or they might be added to both the individual string or to the strings belonging to a group after the alignment has been completed. This strategy for performing alignments is the one proposed by the Feng-Doolittle progressive multiple alignment algorithm [DEKM98].

Figure 5.8 is an example that illustrates progressive multiple alignment and clustering. For ease of presentation, an identifier has been added in front of each string. The distance matrix that is used by the algorithm is in Table 5.2. The distances are calculated using the edit distance metric and giving a cost of 1 to deletion insertion and a cost of 2 to substitution. In order to be able to compare distances between strings of different length, the distances are normalized between 0 and 1. In the clustering phase, agglomerative average clustering is used. If there is more than one equal choice to do the clustering, the first one in the input is chosen. The alignments are performed as described above. If there are more than one possible sequences of edit operations (i.e. alignments) with the lowest cost between two strings, substitutions are preferred over deletions and insertions and insertions are preferred over deletions.

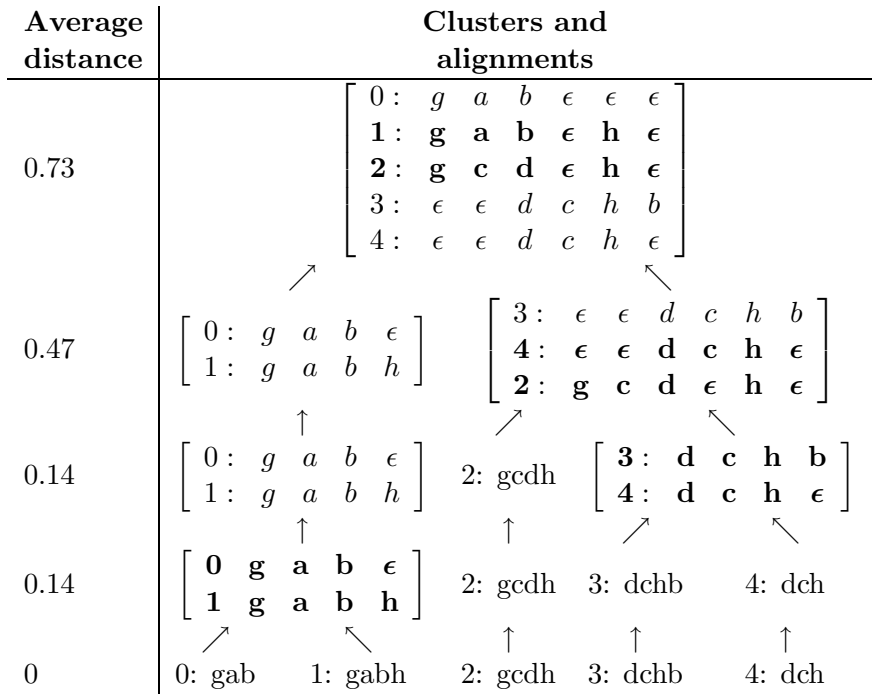


Figure 5.8: Average agglomerative clustering and progressive pairwise alignment for the strings gab, gabh, gcdh, dchb and dch. The clusters are represented by the individual strings and alignments. The pairwise alignment that is performed in each cluster is marked in bold.

	0: gab	**1: gabh**	**2: gcdh**	**3: dchb**	**4: dch**
0: gab	0	0.14	0.71	0.71	1
1: gabh	0.14	0	0.5	0.75	0.71
2: gcdh	0.71	0.5	0	0.5	0.43
3: dchb	0.71	0.75	0.5	0	0.14
4: dch	1	0.71	0.43	0.14	0

Table 5.2: The distance matrix for the example strings gab, gabh, gcdh, dchb and dch.

Chapter 6

Proposed patterns and methods for answer extraction

This chapter describes the novel answer extraction methods. The AEPs along with the concepts related to them are depicted in the first section. The second section first describes the concatenation based answer extraction method and then the alignment based one. The third section details the application of the AEPs. Finally, the scoring of the extracted answer candidates is portrayed. In this thesis, the AEPs are generated from the training data set. All examples and data-specific figures presented in this chapter are drawn from this data. The experimental results presented in Chapter 7 are based on both the training data set and the separate test data set.

6.1 Basic concepts and format of the patterns

Before being able to define the format of the proposed AEPs, the concepts of *token*, *QTag*, *inner context*, *left*, *right* and *leftAndRight context* have to be given. After these six definitions, the format of the AEPs will be described. The last definitions of this section are those of *class specific confidence value* and of *entropy based confidence value*. They are used to score the AEPs.

Definition 6.1 (POS tag) A POS (Part-Of-Speech) tag must contain at least two characters and its capitalization pattern is one of the following: 1) all upper case characters, e.g. NOUN, 2) all lower case characters, e.g. noun, or 3) an upper case character only as the first character, e.g. Noun, depending on the capitalization pattern of the natural language word that it replaces (e.g. UNITA, computer, John, respectively).

The POS tags used correspond to the tag set of 16 tags that the Connexor [1] parser [JT97] uses. All tags are mapped to a variant that contain at least two letters. Only the small case variants are listed in the following. The tags for the open word classes are: abbr (abbreviation), adj (adjective), adv (adverb), en (past participle, e.g. *integrated*), ing (present participle, e.g. *singing*), interj (interjection, e.g. Hey), noun and verb. The tags for the closed word classes are: cc (coordinating conjunction, e.g. *and*), cs (subordinating conjunction, e.g. *if*), det (determiner), inf (infinitive marker, e.g. *to* sing), neg (negative particle, e.g. *not*), num (numeral), prep (preposition), pron (pronoun).

Definition 6.2 (token) A token is a sequence of one or more characters that does not include any whitespace characters. It can be a POS tag of an open class word, a natural language word belonging to a closed word class or a punctuation symbol. An exception to this rule is formed by the numerals, which are closed class words, but which are represented by their POS tag.

We interpret the definition above so that tokens are treated as characters. This means characters in the sense in which they were defined in Definition 5.1 on page 48. In the training data set, the size of the alphabet, i.e. the number of distinct tokens, is 178. Here is a sample sentence before and after it has been transformed into a set of tokens:

```
The last pact failed in 1992 when UNITA, the National Union for
the Total Independence of Angola, lost multi-party elections
and returned to war.
```

```
The last noun verb in num when NOUN , the Adj Noun for
the Adj Noun of Noun , verb adj noun
and verb to noun .
```

Definition 6.3 (QTag) A QTag is a question tag. A question tag is of the form $QWORD_1, QWORD_2, \ldots QWORD_n$, where n is the number of content words (i.e. words not belonging to the set of stop words) in a question. In the word sequence that contains the answer to a specific question, words that occur in the question are replaced with QTags.

Let us illustrate this definition with an example. Below is an example question the answer of which is *the National Union for the Total Independence of Angola*:

[1] http://www.connexor.com

```
What is UNITA?
```

UNITA is the only question word that is not a stop word. Let us take the example from the previous definition to illustrate how the $QWORD$ is replaced with a QTag:

```
The last noun verb in num when QWORD1 , the Adj Noun for
the Adj Noun of Noun , verb adj noun
and verb to noun .
```

Definition 6.4 (inner context) The inner context is a token sequence consisting of the answer string.

It is noteworthy that the inner context may not contain any Qtags as it consists only of a sequence of tokens. Tokens have here the meaning given in definition 6.2. The inner context for the sample sentence given to illustrate Definition 6.3 is:

```
the Adj Noun for the Adj Noun of Noun
```

Definition 6.5 (left, right and leftAndRight context) A left context is an ordered set (i.e. sequence) that may consist of tokens and/or QTags. It consists of at most n items to the left of the inner context. A right context is an ordered set that may consist of tokens and/or QTags. It consists of at most m items to the right of the inner context. A leftAndRight context is an ordered set that may consist of tokens and/or QTags. It consists of at most n items to the left and of m items to the right of the inner context. The left, right and leftAndRight contexts cannot cross sentence boundaries and they cannot contain the answer string. The left (right) context is empty if the answer string is at the beginning (end) of a sentence or if there is another occurrence of the answer string just before (after) it.

As we may see from the definitions above, the inner context, left, right and leftAndRight contexts are sequences of tokens. According to Definition 5.1 on page 48, character sequences are equal to strings. In this method, we treat tokens as characters and thus the methods for strings may be applied to the inner context and to the different contexts as well. As we will see in the next section, the alignment based method for developing answer extraction patterns use the multiple alignment method described in Section 5.4.

Figure 6.1 lists the possible context sizes when $n = 4 = m$ and shows how each of them may be composed from a combination of the left and

Size	Contexts
1	L1, R1
2	L2, R2, L1R1
3	L3, R3, L1R2, L2R1
4	L4, R4, L2R2, L1R3, L3R1
5	L2R3, R3L2, L1R4, L4R1
6	L3R3, L2R4, L4R2
7	L3R4, L4R3
8	L4R4

Figure 6.1: The different context sizes when $n = 4 = m$ and how they may be composed of the left (L) and/or right (R) contexts. For instance, L1 means that the size of the left context is one token or QTag and L4R3 means that the size of the left context is 4 tokens or QTags and that the size of the right context is 3 tokens or QTags.

Size	Contexts		
1	,	,	
2	QWORD1 ,	, verb	, (A) ,
3	when QWORD1 ,	, verb adj	, (A) , verb
	QWORD1 , (A) ,		
4	num when QWORD1 ,	, verb adj noun	QWORD1 , (A) , verb
	, (A) , verb adj	when QWORD1 , (A) ,	
5	QWORD1 , (A) , verb adj	when QWORD1 , (A) , verb	
	, (A) , verb adj noun	num when QWORD , (A) ,	
6	when QWORD1 , (A) , verb adj	QWORD1 , (A) , verb adj noun	
	num when QWORD1 , (A) , verb		
7	when QWORD1 , (A) , verb adj noun		
	num when QWORD1 , (A) , verb adj		
8	num when QWORD1 , (A) , verb adj noun		

Figure 6.2: The different contexts formed from the example sentence presented after Definition 6.3. The maximum left and right context size is 4 as in Figure 6.1. The order in which the contexts are presented is also the same as their order in Figure 6.1. *(A)* in the contexts is used to mark the slot for the answer, i.e. inner context. It has been added only to increase readability.

right contexts or from only one of them. Figure 6.2 lists example contexts that correspond to the possible context sizes. The example contexts are formed from the example that is given for Definition 6.2.

Definition 6.6 (answer extraction pattern AEP) An AEP is a sim-

plified regular expression that consists of an inner evidence and of either a left, right or leftAndRight context. It is a simplified regular expression, because it may only contain the following operators: | (or) and ? (optionality).

A very simple example of an AEP is:

```
QWORD1 , (the Adj Noun for the Adj Noun of Noun) ,
```

As the reader may observe, the above pattern is formed from the leftAndRight context of the form L2R1 which is given in Figure 6.2 as the last context with the size 3 and of the inner context which is given as an example after Definition 6.4. The inner context is separated from the rest of the AEP by parentheses to increase readability.

Two measures for estimating the quality of the AEPs are devised. They are presented inx Definitions 6.7 and 6.8.

Definition 6.7 (class specific confidence value, c(AEP, class)) $c(AEP, class) = |AEP\_in\_Class|/|AEP|$, where $|AEP\_in\_Class|$ is the number of occurrences of the AEP in the question class and $|AEP|$ is the total number of occurrences of the AEP in the data.

As can be seen in the definition, the class-specific confidence value varies between 0 and 1. The more confident the system is with regard to a pattern belonging to a class, the higher the value is. If the value is 1, all instances of the pattern in question belong to the same class. Table 6.3 on page 67 lists information on the class-specific confidence values of the AEPs in the training data. We can observe that most patterns belong to only one class as the median confidence value of the AEPs in all classes is 1.

As some patterns may occur in several classes, another confidence value for them is also used. This confidence value parts from the assumption that the AEP is good if it occurs only in one class and not so good if it occurs in all classes. This is achieved by calculating the **entropy impurity** of the AEP, denoted $i(AEP)$. The definition of $i(AEP)$ is given in Equation 6.1.

$$i(AEP) = -\sum_{j \in Classes} Items(j) log_2 Items(j), \tag{6.1}$$

where $Items(j)$ is the proportion of the items in the set of similar AEP that belong to the class j. The value of $i(AEP)$ is 0 if all instances of the same AEP belong to the same class. The greatest value for $i(AEP)$ is obtained when the items of AEP are equally distributed in all classes. The value of $i(AEP)$ is then $log_2(|classes|)$, where $|classes|$ is the number of classes.

When the number of possible classes is 8, the maximum entropy impurity value is 3. Table 6.2 on page 65 lists information on the entropy impurity values of the AEPs in the training data. We can observe that most patterns belong to only one class as the median entropy impurity value of patterns of all lengths is 0. In order to use $i(AEP)$ as an additional confidence value, let us call it the entropy based confidence value (c_i), we will scale it between 0 and 1 and we will subtract it from 1 so that the maximum c_i reflects a pattern we are confident with and the minimum c_i reflects a pattern we are not confident with. More formally, $c_i(AEP)$ is calculated as shown in Definition 6.8.

Definition 6.8 (Entropy based confidence value, $c_i(AEP)$)

$c_i(AEP) = 1 - \frac{i(AEP)}{max}$, where max is the maximum entropy value given the number of classes.

6.2 Pattern generation methods

Now that we know what the AEPs are composed of, we are ready to proceed to examine how they can be generated. The first subsection of this section presents the concatenation based method for generating AEPs and the second subsection presents the alignment based method.

6.2.1 Concatenation based

The concatenation based method for forming AEPs serves as a baseline against which the alignment based method that will be described in 6.2.2 is compared. The concatenation based method for forming AEPs takes as input a file containing classified natural language questions, each followed by at least one text snippet into which an answer string to the question is marked. As output the method produces a set of concatenation based AEPs (CAEPs) for each class. The method consists of the following three steps:

1. For each occurrence of an answer string, form the inner context and the left, leftAndRight and right contexts. Store each context in a class-specific set.

2. For each class-specific set of contexts:

   (a) Prune duplicates away.

(b) For each inner context: Concatenate it with each left, leftAndRight and right context to form the CAEPs.

3. Calculate both the class-specific confidence value and the entropy based confidence value for each CAEP as explained in Definition 6.7 (on page 61) and in Definition 6.8 (on page 62), respectively.

The total number of class-specific CAEPs is |contexts| ∗ |inner|, where |contexts| is the number of unique left, right and leftAndRight contexts and |inner| is the number of unique inner contexts. In order to obtain a rough estimate of the cardinality of the class-specific groups of CAEPs, let us calculate the hypothetic example cardinalities for the classes *Other* and *Person Definition.*

In step *1* of the algorithm, we extract the context of every occurrence of every answer string from the text snippets and form the left, leftAndRight and right contexts. Let there be two answer strings in each text snippet retrieved. According to Table 6.1, this would mean 44 $(2 * 22)$ answer strings in the class *Other* and 168 $(2 * 84$ answer strings in the class *Person Definition*. Now, if the maximum size of the left and right contexts is 4, which is the case in Figure 6.1 on page 60 and in the instantiation of the algorithm presented in this thesis, the number of different contexts is 24. The number of contexts obtained for the class *Other* is 1056 $(24 * 44)$ and for the class *Person Definition* it is 4032 $(24 * 168)$. These contexts are transformed into the format given in Definition 6.5 on page 59. The inner contexts are formed simply by processing the answer strings into the format given in Definition 6.4 on page 59.

In step *2(a)* of the algorithm, the duplicate contexts inside each class are first pruned away. Let 20% of the contexts be duplicates. Now the number of contexts in the class *Other* is roughly 844 and in the class *Person Definition* roughly 3226.

In step *2(b)*, each unique inner context is concatenated with each left, leftAndRight and right context. Let there be one unique inner evidence per question. In the training data, the number of non-NIL questions in the class *Other* is 15 (see Table 4.2 on page 35).Thus, the number of CAEPs produced for the class *Other* is 12660 $(15 * 844)$. Let us now roughly estimate a possible cardinality of the group of CAEPs in the class *Person Definition.* In the training data, the number of non-NIL questions in the class is 24 (see Table 4.2 on page 35).Thus, the number of CAEPs produced for the class *Person Definition* is 48390 $(15 * 3226)$. These hypothetic example estimates will be compared with real figures later on in this section.

The method described above is applied to the training data. The questions and their classes are given in Appendix 1. The text snippets con-

taining the answers are extracted from the newspaper corpora. Only those documents are considered that match the query containing all the content words of the question and the answer string. This explains why some question answer pairs produce no text snippets at all even though all non-NIL questions in the training data do contain an answer in the text collection. Table 6.1 illustrates the number of answer snippets that the questions of the training data produce. The figures are given for all classes together and for each class separately. For each class, the average, minimum and maximum number of answer text snippets produced is given. In addition, the number of questions not producing any answer text snippets is given.

Class	**Avg**	**Min**	**# of zeros**	**Max**	**# of snippets**
Location	5.29	0	4	24	111
Measure	1.35	0	10	11	27
Organization	3.93	0	3	25	55
Organization D	1.67	0	1	7	35
Other	1.47	0	4	5	22
Person	4.59	0	6	68	133
Person D	3.50	1	-	19	84
Time	3.63	0	3	13	58
All	3.28	0	31	68	525

Table 6.1: The overall and class-specific average, minimum and maximum numbers of answer text snippets that are produced by the question answer pairs. The number of pairs that result in no answer text snippets at all is also given as well as the absolute number of answer text snippets obtained.

Table 6.2 shows how the number of unique CAEPs and their entropy impurity value varies as the size of the left, right and leftAndRight contexts varies. The number of unique CAEPs is greatest when the context size is 4. This might be because the number of different ways in which the context can be formed is greatest when the context size is 4, as was illustrated in Figure 6.1 on page 60. The number of different CAEPs is smallest when the context size is 1. This holds even though there are two different ways (L1 and R1) in which the contexts of the size of 1 can be formed and only one single way (L4R4) in which the contexts of size 8 can be formed. This might be due to the fact that when the context size is short, there are more similar contexts (and thus also more similar CAEPs), whereas the longer the context size is, the more there are different contexts.

The table also shows that the entropy impurity value is low when the CAEP contexts are long and that it grows as they shorten. The change

in entropy impurity value according to context size can be used as a reliability indicator in determining the maximum context size. For example, in Table 6.2 we can observe that there is no difference in the minimum, maximum, median, mean or variance of the entropy impurity values when the context size is grown from 7 to 8. This suggests that the CAEPs with context size 7 might be as reliable as those of size 8. As the patterns with context size 7 are more general than those having a context of the size of 8, there is no need to use the longer patterns.

Size	**#**	**%**	**min**	**max**	**median**	**mean**	**var**
Any	164964	100	0.000	2.4710	0.000	0.01281	0.014130
1	**2639**	**1.6**	0.000	**2.4710**	0.000	**0.12550**	**0.143675**
2	11547	7.0	0.000	2.2000	0.000	0.06307	0.067560
3	23425	14.2	0.000	2.0000	0.000	0.02442	0.025344
4	**36952**	**22.4**	0.000	2.0000	0.000	0.00769	0.007990
5	33983	20.6	0.000	2.0000	0.000	0.00334	0.003528
6	27384	16.6	0.000	1.0000	0.000	0.00179	0.001769
7	19136	11.6	0.000	1.0000	0.000	**0.00096**	**0.000941**
8	9898	6.0	0.000	1.0000	0.000	**0.00096**	**0.000941**

Table 6.2: Information on how the entropy values differ as the size of the context is altered in the dataset consisting of CAEPs.

Table 6.3 illustrates the class-specific distribution of the confidence values of the CAEPs for the data. The table also lists the number of CAEPs in each class. As we can see in the table, the number of CAEPs varies in different classes. For example, in the training data, 50 % of all patterns belong to the class *Definition Person*, and only 1 % are in the class *Other* and only 2% in the class *Measure*. When we hypothetically estimated the number of *Definition Person* CAEPs, we arrived at the number 48390 which is not so much less than a half of 83386, the actual number. For the class *Other*, the hypothetical number was 12660, which is about five times greater than the actual number, 2505. From these figures we may infer that the class *Person Definition* may contain more than 2 answer strings per text snippet on average and that it may contain less than 20% of similar contexts. The number of unique inner context cannot be greater than the number of questions – which was used in calculating the hypothetical number of CAEPs – because in the training data used, each question has only one right answer. For the class *Other*, we may infer that the text snippets might contain only one answer string, which is the minimum, and that it may contain more than 20% of similar contexts. Also the number

of unique inner contexts may be smaller than the number of questions as some inner contexts may become similar after preprocessing. However, a closer study would be needed in order to be able to tell exactly why the number of CAEPs is considerably lower for the class *Other* than for the class *Person Definition*.

In general, we can say that those classes that have small numbers of CAEPs (i.e. the classes *Other* and *Measure*) also have a small number of text snippets from which the CAEPs have been generated as can be observed in Table 6.1 on page 64. However, the number of class-specific CAEPs does not seem to affect the mean confidence value of the class, as both *Definition Person* and *Measure* have an average value which is above the average value of all classes. On the other hand, the mean confidence value of the CAEPs of the class *Other* is lower than the mean value of all classes. In Table 6.3 we can further observe that there are three classes where the average mean confidence value of the CAEPs is higher than the average mean of all classes. These classes are: *Measure*, *Definition Person* and *Time*. Incidentally, the variance of the values of these classes is also lower than the variance of all classes. We could conclude that the classification adopted suits these classes better than the others when the training data is used. However, the differences in confidence values between classes are very small.

The total number of CAEPs differs in the table illustrating the difference in entropy impurity as the context size varies (Table 6.2) and in the table illustrating the difference in confidence values in different classes (Table 6.3). In Table 6.2, the total number of CAEPs is 196 176 and in Table 6.3, it is 199 293. This is because some patterns belong to several classes, and the same pattern is thus taken into account several times in Table 6.3.

6.2.2 Alignment based

The alignment based method for forming AEPs takes as input the CAEPs produced by the concatenation based method. As output the method produces a set of generalized CAEPs (GAEPs). The generalizations are expressed as CAEPs that contain the operators | (or) and ? (optionality). The GAEPs are obtained by performing an agglomerative hierarchical clustering with pairwise alignment to the CAEPs. All processing of CAEPs is performed in class- and maximum QTag-specific sets. This means that the CAEPs are first divided into class-specific groups and then into maximum QTag-specific groups. The maximum QTag of a CAEP is the $QWORD_n$ where n is largest. For example, in the following CAEP

Class	#	%	min	max	median	mean	var
ALL	167646	100	0.0051	1.0000	1.0000	0.9844	0.0096
LOC	20823	12	0.0175	1.0000	1.0000	0.9723	0.0166
MEA	3375	2	0.0217	1.0000	1.0000	**0.9959**	**0.0026**
ORG	11603	7	0.0112	1.0000	1.0000	0.9685	0.0171
ORGD	14338	9	0.0154	1.0000	1.0000	0.9747	0.0177
OTH	**2505**	1	0.0094	1.0000	1.0000	**0.9582**	**0.0291**
PER	20013	12	**0.0051**	1.0000	1.0000	0.9789	0.0127
PERD	**83386**	50	**0.0435**	1.0000	1.0000	0.9926	0.0043
TIM	11603	7	0.0139	1.0000	1.0000	0.9908	0.0059

Table 6.3: Class-specific information on the number of CAEPs and on their confidence values.

```
QWORD7 verb QWORD4 to (Noun Noun Noun and Noun)
```

the maximum QTag is *QWORD7*. The class- and maximum QTag-specific groups and their cardinalities are listed in Tables 6.4 and 6.5 on the rows denoted by C. The groups are obtained by grouping the CAEPs of the training data and they constitute the initial situation before any clustering and alignments is done. As can be seen in the table, only patterns containing QTags are clustered. Those without any QTags are left as such. The class-specific numbers of patterns not containing any QTags are the following: LOC: 17296, MEA: 2335, ORG: 10547, ORGD: 9634, OTH: 1410, PER: 10067, PERD: 34759 TIM: 5975.

Each of the groups is clustered using agglomerative hierarchical complete link clustering. If the size of the group contains too many items for the clustering algorithm and the computer used, the group has to be split. In our case, the group *Person Definition* with two QTags that contained 38718 CAEPs had to be split into four parts. If the number of items in a group is greater than 49, clustering is applied to it so that the desired number of clusters is set to $\lceil |n|/20 \rceil$, where $|n|$ is the number of items in the cluster. If the result of this clustering still contains clusters (or groups) of 50 or more items, those groups are clustered again in the same way. The clusterings are repeated until no cluster contains over 49 items. At this phase, the final clustering with MSA is performed. In the final clustering, the desired number of clusters is the same as in the previous ones, i.e. $\lceil |n|/20 \rceil$.

In order to be able to cluster a group of CAEPs, a dissimilarity matrix containing the dissimilarities between them has to be created. Such a matrix and its use in agglomerative hierarchical clustering was illustrated in

QTag	C/A	Class			
		LOC	MEA	ORG	ORGD
1	C	1248	135	354	4184
	A	5641	384	2704	412014
2	C	1764	505	354	220
	A	56281	13288	1164	482
3	C	189	150	150	-
	A	349	609	240	-
4	C	317	100	144	300
	A	4169	337	437	654
5	C	-	100	18	-
	A	-	400	24	-
6	C	-	-	**6**	-
	A	-	-	**6**	-
7	C	9	50	-	-
	A	22	60	-	-
8	C	-	-	24	-
	A	-	-	80	-
9	C	-	-	**6**	-
	A	-	-	**6**	-
Total	C	3527	1040	1056	4704
	A	66462	15078	45545	413150

Table 6.4: The number of class- and maximum QTag-specific CAEPs followed by the corresponding number of CAEPs that the GAEPs would produce. Only information concerning the patterns for the classes LOC, MEA, ORD and ORGD are given. The rest of the classes are presented in Table 6.5.

Section 5.2. In order to create the dissimilarity matrix, in its turn, a dissimilarity measure for measuring the dissimilarity between two CAEPs has to be devised. The dissimilarity measure adopted in this work is the edit distance [WF74], which was described in Section 5.3. If the edit distance between two CAEPs is important, they are regarded as very dissimilar, and if it is small, they are regarded as very similar. As we saw in Section 5.3, there exist many different variations of the basic edit distance metric. The one used in the method for creating the GAEPs is a variant of the basic edit distance where the cost of the insertion and deletion operations is one and the cost of the substitution operation is two. Let us call this variant of the basic edit distance *edit distance for alignment* and denote it with D'.

QTag	C/A	Class			
		OTH	PER	PERD	TIM
1	C	479	744	8964	1925
	A	257390	1742	1446914	102559
2	C	182	3378	**38718**	1036
	A	340	102133	**8500550**	250072
3	C	126	3528	945	168
	A	267	36213	1848	473
4	C	308	2000	-	1176
	A	685	79163	-	140322
5	C	-	80	-	903
	A	-	180	-	20775
6	C	-	96	-	420
	A	-	352	-	74224
7	C	-	120	-	-
	A	-	383	-	-
8	C	-	-	-	-
	A	-	-	-	-
9	C	-	-	-	-
	A	-	-	-	-
Total	C	1095	9946	48627	5628
	A	258682	220166	9949312	588425

Table 6.5: The number of class- and maximum QTag-specific CAEPs followed by the corresponding number of CAEPs that the GAEPs would produce. Only information concerning the patterns for the classes OTH, PER, PERD and TIM are given. The rest of the classes are presented in Table 6.4.

The cost function for D' is thus:

$$c(s_i \rightarrow \epsilon) = 1$$

$$c(\epsilon \rightarrow t_j) = 1$$

$$c(s_i \rightarrow t_j) = \begin{cases} 0 & if \quad s_i = t_j \\ 2 & otherwise \end{cases}$$

The notations used in the above cost function definition are the same as those used in Section 5.3 that introduces the basic edit distance and its cost function, among others. This setting makes the metric close to the insertion deletion edit distance, where the cost of the substitution operation is greater than 2 and the costs of the insertion and deletion operations are 1.

However, in our setting, the cost of a substitution operation is always the same as that of one insertion and one deletion operation. This means that typically several equally expensive alignments are produced for a pair of CAEPs. In order to choose one from these alignments, the method prefers substitution over deletion and insertion, and insertion over deletion. This choice was taken because preferring substitution over insertion and deletion makes the generated regular expressions less general. The MSA based method for creating regular expressions makes a considerable number of generalizations so we are careful not to introduce too many generalizations in the other parts of the method. This point will be explained in detail at the end of this section where the creation of regular expressions from alignments is explained. The preference of insertion over deletion in choosing one alignment from a set of alignments having equal costs is completely arbitrary. In fact, as will be seen at the end of this chapter, preferring deletion over insertion would produce exactly the same regular expression.

In order to be able to produce a dissimilarity matrix with comparable dissimilarity values, the edit distances are normalized between 0 and 1. In the case of the edit distance for alignment D', normalization is achieved by the following calculation:

$$D'_{normalized}(A, B) = \frac{D'(A, B)}{|A| + |B|},$$

where A and B are CAEPs, $|A|$ and $|B|$ are the lengths of the corresponding CAEPs.

The procedures of performing the complete link agglomerative hierarchical clustering and alignment were described in Sections 5.2 and 5.3. Agglomerative hierarchical clustering was chosen over the other clustering methods and over divisive hierarchical clustering because MSA can be performed at the same time as the clustering. Complete link clustering was chosen because the groups of CAEPs to be clustered are typically very heterogeneous: there may be one CAEP that is completely different from the others and a large cluster of CAEPs that are very similar to each other. Complete link clustering keeps the very different CAEPs away from the same clusters as it always merges those clusters whose most different CAEPs are the most similar. Having very different CAEPs in the same cluster would produce useless regular expressions.

The alignments are used to produce simplified regular expressions that can be used as AEPs. As a regular expression produced through alignment typically matches more patterns than the single CAEPs from which it has been derived, it is called a generalized CAEP (GAEP). Producing simplified regular expressions from alignments is straightforward. The alignments are

read into a matrix. For each column, an expression is printed. There are four different possibilities:

1. If a column only consists of occurrences of the same token or QTag, print the token or QTag, accordingly.

2. Else if a column contains one or more occurrences of the same token (or QTag) and the deletion or insertion symbol 'X', print a symbol denoting optionality after the token: token? (or after the QTag: QTag?).

3. If a column contains different tokens and/or QTags, print them as a disjunction: $\text{tokenOrQTag}_1|\text{tokenOrQTag}_2|\ldots|\text{tokenOrQTag}_n$.

4. If a column contains different tokens and/or QTags and the symbol 'X', print the tokens and/or QTags as an optional disjunction: $(\text{tokenOrQTag}_1|\text{tokenOrQTag}_2|\ldots|\text{tokenOrQTag}_n)?$.

QWORD7	verb	QWORD4	to	(Noun	Noun	Noun	and	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	X	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	and	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	on	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	X	noun)

Figure 6.3: An example alignment from which a GAEP is produced. The inner evidence is surrounded by parenthesis for clarity.

Figure 6.4 shows an example of an alignment from which the regular expression below - or GAEP - is produced. The inner context of the GAEP is surrounded by <answer> tags.

```
QWORD7 verb QWORD4 to <answer>Noun (Noun)? (Noun)? (and|on)?
(Noun|noun)</answer>
```

The 13 sequences that the produced GAEP matches in addition to the sequences from which it was generated through alignment are listed in Figure 6.5

The MSA based generation of AEPs produces regular expressions that are quite accepting because the character *X* is introduced to the alignments in order to fill any gaps. The *X*s, in their turn, are translated into the operator ? which denotes optionality and makes the regular expressions very accepting. In order to keep the amount of ? operators originating from the alignment produced while calculating the edit distance to its minimum,

QWORD7	verb	QWORD4	to	(Noun	Noun	Noun	and	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	X	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	and	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	on	Noun)
QWORD7	verb	QWORD4	to	(Noun	X	X	X	noun)

Figure 6.4: An example alignment from which a GAEP is produced. The inner evidence is surrounded by parenthesis for clarity.

QWORD7	verb	QWORD4	to	(Noun	Noun Noun and noun)
QWORD7	verb	QWORD4	to	(Noun	Noun Noun on Noun)
QWORD7	verb	QWORD4	to	(Noun	Noun Noun on noun)
QWORD7	verb	QWORD4	to	(Noun	Noun and Noun)
QWORD7	verb	QWORD4	to	(Noun	Noun and noun
QWORD7	verb	QWORD4	to	(Noun	Noun on Noun)
QWORD7	verb	QWORD4	to	(Noun	Noun on noun)
QWORD7	verb	QWORD4	to	(Noun	and noun)
QWORD7	verb	QWORD4	to	(Noun	on noun
QWORD7	verb	QWORD4	to	(Noun	Noun Noun Noun)
QWORD7	verb	QWORD4	to	(Noun	Noun Noun noun)
QWORD7	verb	QWORD4	to	(Noun	Noun Noun)
QWORD7	verb	QWORD4	to	(Noun	Noun noun

Figure 6.5: The 13 sequences that the GAEP matches in addition to the 5 sequences listed in Figure 6.4 from which it was produced. The inner evidence is surrounded by parenthesis.

substitution is preferred over insertion and deletion. Substitution produces regular expressions that are less accepting than those produced by insertion and deletion because substitutions are transformed into the | operator. Tables 6.4 and 6.5 illustrate the number of class- and maximum QTag-specific CAEPs and the corresponding number of CAEPs that the GAEPs would produce. In the tables, the C stands for concatenation and the A for alignment. The CAEPs are produced by the concatenation based method and the CAEPs that the GAEPs produce are generated by the alignment based method. For example, the concatenation based method has generated 1248 CAEPs with a maximum QTag of 1 for the class location. The alignment based method has generated GAEPs with the maximum QTag of 1 that can be expressed as 5641 distinct CAEPs for the class location. Table 6.4 presents the numbers of distinct CAEPs for the classes LOC, MEA, ORG and ORGD. Table 6.5 presents the numbers of distinct CAEPs for the rest

of the CAEPs, i.e. those CAEPs that belong to the classes OTH, PER, PERD and TIM.

6.3 Application of the patterns

This section first describes how the proposed AEPs are used in a QA system in order to extract answer candidates from text. The second subsection describes how the extracted answer candidates are scored in order to be able to select the best answer.

6.3.1 Answer candidate extraction in text

The application of the AEPs means that they are matched to the output of the *Paragraph Selector and Preprocessor* in order to find the answer. The software module that performs this is called the *Answer Extractor*, and it is illustrated in Figure 4.1 on page 28 among the other modules of the QA system. As the AEPs are class-specific, only the patterns belonging to the class of the question are matched. In addition to this, only patterns whose greatest QTag is smaller or equal to the number of question words, are applied.

Let us illustrate the pattern selection, instantiation and matching using the following question:

```
 PERSON F Who was Haiti's former military commander in
chief?
```

For the example question above, the patterns belonging to the class *Factoid Person* and having a maximum QTag of 4, 3, 2 and 1 or not having any QTag at all, are matched. This is because the class of the question is *Factoid Person* and because the number of content words (i.e. words not belonging to the stop word list) is 4. The order of the content words in the question determines which QTags they replace. In our example question, they are as follows:

```
question word 1: Haiti's   , replaces QWORD1
question word 2: military  , replaces QWORD2
question word 3: commander , replaces QWORD3
question word 4: chief     , replaces QWORD4
```

When performing answer extraction for the example question, one of the patterns to be instantiated is:

```
QWORD1 Adjective QWORD2 QWORD3 in QWORD4 COMMA <answer>Abbr
Abbr Noun Noun</answer>
```

After instantiation the pattern becomes:

```
Haiti's Adjective military commander in chief COMMA <answer>Abbr
Abbr Noun Noun</answer>
```

The instantiated pattern matches a text snippet such as:

```
Haiti's former military commander in chief, Lt. Gen. Raoul
Cedras
```

The example pattern extracts from the text snippet the correct answer for the question, which is

```
Lt. Gen. Raoul Cedras
```

6.3.2 Answer candidate scoring

The matching process typically produces many answer candidates among which the answer is chosen. The selection is performed by composing a score reflecting the confidence the system has in the correctness of the answer. The answer with the highest confidence score is chosen. Thus, the scoring of an answer candidate plays a central role in the method.

The score of an answer candidate is dependent on the five following properties: the document similarity (*DocSim*) (see Section 4.1 beginning on page 27), the length of the answer candidate (*Length*), the length of the context of the AEP that extracted the answer candidate (*Context*), the number of QTags in the AEP that extracted the answer candidate (*QTags*) and the frequency of the answer candidate (*Freq*). These properties were chosen by manually inspecting the the training data in order to figure out properties that might best differentiate between correct and wrong answer candidates. All five properties are normalized so that their maximum value is 1. The properties *DocSim* and *Context* are global in the sense that their values are comparable across different questions. *DocSim* always ranges from 0.4 to 1 and *Context* always ranges from 1/8 to 1. The other values are question specific, which means that their values are only comparable across answer candidates given to the same question. As the score is only used to compare the answer candidates of a question with each other, question-specific properties are adequate. If a threshold score was used to determine *NIL* questions or to determine when a sufficiently good answer has been

found, global properties would be more important. The five properties used in the score are described with more detail in the following.

DocSim is calculated for entire documents only and thus it is the same for all paragraphs belonging to one document. The system has a document similarity value threshold, which means that documents with a document similarity value smaller than the threshold value of 0.4 are not processed at all. This not only speeds up the processing, but it is also used to determine *NIL* questions. How the document similarity threshold value was determined was explained at the end of Section 4.1, which begins on page 27.

The property *Length* is the number of tokens that the answer candidate contains. It is normalized by dividing it by the maximum *Length* among all answer candidates for the question. This property may take a value between $1/n$ and 1, where n is the question-specific maximum *Length*.

The property *Context* is actually the average of all the lengths of the AEP contexts that have extracted the same answer candidate. The length of an AEP context is the number of tokens and QTags in the left and right context of the AEP. The length of an AEP context is normalized by dividing it by 8, which is the global maximum length of an AEP context. Thus, the value of this property ranges between 1/8 and 1.

The property *QTags* is the average of the number of QTags in the AEPs that have extracted the answer candidate. The number of QTags is normalized by the maximum number of QTags in an AEP that has matched while searching for an answer to the question. The value of *QTags* ranges between 0 and 1.

The term *Freq* is the number of times an AEP or several distinct AEPs have found the same answer candidate. This term is normalized by dividing it by the question-specific maximum *Freq*. The term *Freq* exploits the redundancy of answer candidates in the text corpus. This feature has been successfully exploited in several QA systems, such as the QA system of the MultiText Group [CCK+02] that has participated in several TREC campaigns and the Web based QA system of Dumais et al. [DBB+02].

The properties described above are weighted equally and compiled into one common score as shown in Equation 6.2.

$$Score(AC) = \frac{DocSim_{AC} + Length_{AC} + Context_{AC} + QTags_{AC} + Freq_{AC}}{5} \tag{6.2}$$

The answer candidate AC with the highest score is chosen to be the answer A to the question. This is illustrated in Equation 6.3.

$$A = max_{AC \in ACS} Score(AC), \tag{6.3}$$

where ACS is the set of answer candidates. The properties of Equation 6.2 are weighted equally because manual inspection of the training data suggested that they would all be equally important. However, a close inspection of the results of the QA system might suggest a more fine-grained weighting. In the next chapter, which describes the experimental results, Tables 7.3, 7.4, 7.7, 7.8, 7.11, 7.12, 7.15, 7.16, (on pages 82, 83, 86, 86, 88, 88, 91, 91, respectively), present detailed information on the values of the properties *DocSim*, *Context* and *QTags* for correct and wrong answer candidates. The significance of these results will be discussed in Section 8.1, beginning on page 101.

Chapter 7

Experimental results

The previous chapter detailed the novel AEPs and the two novel methods – the concatenation and alignment based – for generating them. The necessary background techniques for understanding these methods was given in Chapter 5. Chapter 4 described the training and test data as well as the QA system inside which the new AEPs are incorporated. Thus, we now have all the necessary information to proceed on to describe the experimental setting and the results of the experiments.

This chapter briefly first describes the experimental setting for evaluating the concatenation and alignment based answer extraction methods and then presents the results. Detailed results of the experiments with the concatenation based method are presented in the second section. The third section contains the results of experiments with the alignment based method. Section four presents a summary and comparison of the results as well as some side products that the methods produce. Discussion about the results and their significance will be in the next chapter.

7.1 Description of the experimental setting

The experiments whose results are presented in the tables of the next two sections consist in testing both the concatenation based method and the alignment based method. The methods are tested both with the training data and with the test data. Performing experiments with the training data shows how well the method is fitted to the specific data from which the patterns are induced. The hypothesis is that the concatenation based method that performs less generalization than the alignment based one would perform considerably better on training data than on test data. On the other hand, the hypothesis is that the alignment based method would

perform worse than the more simple concatenation based method on the training data but better on the test data. However, the experimental results presented in the following do not confirm this hypothesis. This and the reasons behind it is discussed at more length in Section 8.1. The degree of difference between the datasets also contributes to the results. If the training and test data are very similar, the AEPs learned from the training data would apply very well to the test data.

7.2 Concatenation based method

This section presents the results of the experiments with the concatenation based method. The results are first given from experiments using the training data (runs ending in *Train*, e.g. $concTrain_1$) and then using the test data (runs ending in *Test* e.g. $concTest_1$). For both datasets, two types of result tables are presented. The first type of tables contain information on the correctness of the answers given by the system. Examples of this kind of tables are Tables 7.1 and 7.2. The second table type gives detailed information on the number of processed text fragments and on the matched AEPs. Example of such tables are Tables 7.3 and 7.4.

7.2.1 Training data

The following tables present results using the training data. The baseline for the training data set is $20/180 \approx 11.1\%$, which is achieved by returning *NIL* as an answer for every question.

Tables 7.1 and 7.2 present information on the correctness of the answers. Table 7.1 contains information about the performance of the system both on the entire question set of 180 questions and on the factoid (*F*), definition (*D*) and NIL questions separately. The number of questions in each dataset is marked in square brackets in the table. The factoid and definition question datasets do not include the NIL questions even though also they are categorized into these two classes as can be observed in the Appendices 1 and 2. This is because the performance of the system with regard the NIL and non-NIL questions is quite different and thus mixing results concerning both datasets would not be very informative.

The results are given for two runs: $concatTrain_1$ and $concatTrain_3$. The indices *1* and *3* at the end of the runs denote the maximum number of answer candidates among which the answer is chosen. If the index is *1*, only the best scoring answer candidate is considered, and if it is *3*, the three best scoring answer candidates are considered. As can be seen from the table, the percentage of correct answers is considerably higher when

taking into account the three first answers than when taking into account only the first answer: 56.6% versus 44.4%. The answer rank score (*AR Score* in the table) is 0.333 for both runs. The answer rank score presented in the table is the average of the question-specific scores. The question-specific score itself is the reciprocal of the rank of the first correct answer in the list of answers returned by the system. If the system does not return any correct answer or if it returns *NIL* for a non-NIL question the question is given the score 0. The AR Score is naturally equal to both of the runs presented in the tables because it takes into account all answers given by the system and not only the first or first three answers. Precision (P), recall (R) and F_1-measure (F_1) are calculated for the NIL questions. They are 0.308, 0.800 and 0.444, respectively.

run	**Right** % [180]	**Wrong** % [180]	**X** % [180]	**AR Score** [180]	**Right** % F [115]	**Right** % D [45]	**NIL** [20] **P**	**R**	$\mathbf{F_1}$
concatTrain$_1$	44.4	51.1	4.4	0.333	36.5	48.9	0.308	0.800	0.444
concatTrain$_3$	50.6	42.2	7.2	0.333	40.0	62.2	0.308	0.800	0.444

Table 7.1: Results of the concatenation based method with training data. All questions are considered. The symbol X stands for inexact answers.

Table 7.2 presents the class-specific information on the performance of the QA system when it uses the concatenation based method and the training data. The classes are marked by the same mnemonic abbreviations that are used throughout the thesis. They are explained in Section 4.2 on page 30. As can be observed in the table, the best results – 70.8% of the answers are correct – are obtained for the class of *Definition Person* questions in the run where the three first answers are taken into account. The worst results are obtained for the class of *Other* questions where only 26.7% of the answers are correct in the run that takes into account only the first answer.

Table 7.3 contains detailed information on the number of processed text fragments and on the matched answer extraction patterns. The title fields of the table are: *Type*, *Frag*, *ACand*, *DocSim*, *Context* and *QTags*. The four last ones are divided into the categories *Right* and *Wrong*. Each row in the tables represents summarized information for a certain kind of question. The title field *Type* designates the category of the question for which the subsequent information is given. The reader may have observed that the *types* consist of class names or of the word *ALL* concatenated with the character *W* or *R*. The last character naturally tells whether the system returned a *W*rong or *R*ight answer to the question. Thus, for example the

run	*Correct Answers %*								
	Definition		*Factoid*						*Both*
	ORG [21]	**PER** [24]	**LOC** [21]	**MEA** [20]	**ORG** [14]	**OTH** [15]	**PER** [29]	**TIM** [16]	[160]
concatTrain$_1$	42.9	**54.2**	38.1	40.0	50.0	**26.7**	34.5	31.3	40.0
concatTrain$_3$	52.4	**70.8**	42.9	40.0	50.0	**33.3**	37.9	43.8	46.9

Table 7.2: Percentage of correct answers for factoid and definition questions provided by the concatenation based method with training data. Breakdown according to answer type. Only answers for non-NIL questions are presented.

row labeled with *LOCW* shows summarized information for all questions whose class is *Location* and for which the system returned a wrong answer. Here the question is classified as being right only if the correct answer is at the first rank in the list of returned answers. Inexact answers are regarded as wrong. The next title field is *Frag*. By observing it, we can see that the highest average number of text fragments that the system analyzed in order to produce the answer is 3863.0. This figure was obtained for questions of class *Person* for which the system produced a right answer. The lowest average number of text fragments observed is 60.4, and it was obtained for the questions belonging to the class *Measure* that were answered wrong by the system.

The rest of the title fields are divided into two parts, *Wrong* and *Right*. This means that figures for wrong and right answer candidates are given separately. It might sound contradictory that a *Type* that contains information about questions that have received a wrong answer have right answer candidates. However, it is not at all contradictory as such questions usually have right answer candidates that for some reason were not chosen to be ranked first on the answer list returned by the system. *ACand* means the total number of answer candidates returned by the system. The total number of text fragments processed is the sum of the wrong and right answer candidates. This is because one answer candidate is picked from each processed text fragment. The same answer candidate typically has several occurrences. For every *Type*, the number of wrong answer candidates produced is considerably higher than the number of right answer candidates. The highest average number of wrong answer candidates produced is 3662.9. It is produced for the questions of class *Definition Person* that are answered correctly. The lowest average number of wrong answer candidates – 59.1 – is produced for questions of class *Measure* that are answered wrong. The highest average number of right answer candidates produced

is 200.1. It is produced for the questions of class *Person* that are answered correctly. The lowest average number of right answer candidates – 0.3 – is produced for questions of class *Organization* that are answered wrong.

DocSim is the average document similarity value (See Section 5.1 on page 41 for the definition.) of all answer candidates produced for the same *Type*. The same answer candidate may have several document similarity values as it may have been extracted from several text fragments. By looking at the average figures produced for all *Types*, i.e. *ALLW* and *ALLR*, we can observe that the average document similarity values are higher for right answer candidates – 0.6244 and 0.6412 – than for wrong answer candidates – 0.6131 and 0.6234, but that the differences are not very important. The highest average document similarity value for wrong answer candidates is 0.7710 and it occurs for questions of class *Measure* that are answered correctly. The lowest average document similarity value for wrong answer candidates is 0.4749 and it occurs for questions belonging to the class *Organization* and that are answered correctly. The highest average document similarity value for right answer candidates is 0.8230 and it occurs for questions of class *Other* that are answered wrong. The lowest average document similarity value for right answer candidates is 0.4710. It occurs for questions belonging to the class *Organization* and that are answered wrong.

Context means the average size of the left and right contexts of the AEPs that have extracted the answers of the *Type*. Naturally also in this case there may be several patterns with varying context sizes that have extracted the same answer candidate. The average context sizes vary from 1.326 (wrong answer candidates for the type *OTHR*) to 4.066 (right answer candidates for the type *OTHW*). In general, for figures concerning the same *Type*, the average context sizes for right answers are always higher than those for wrong answers. The only exception is the *Type ORGW*, where the average context size for wrong answers is 1.601 and for right answers it is 1.500.

The last columns of the table tell the average number of *QTags* (see Definition 6.3 on page 58 for the definition) in the AEPs that have extracted the answers of the *Type*. Naturally also in the case of QTags there may be several patterns with varying numbers of QTags that have extracted the same answer candidate. The average number of QTags varies from 1.011 (wrong answer candidates for the type *ORGW*) to 2.207 (right answer candidates for the type *PERDR*). For the figures concerning the same *Type*, the average number of QTags for right answers is always higher than those for wrong answers.

Detailed information about the processed text fragments and the AEPs

Type	Frag	ACand		DocSim		Context		QTags	
		Wrong	Right	Wrong	Right	Wrong	Right	Wrong	Right
LOCW	3344.3	3317.1	27.25	0.6046	0.6538	1.509	1.933	1.021	1.200
LOCR	1434.9	1347.1	87.7	0.5614	0.6136	1.632	3.557	1.041	1.361
MEAW	**60.4**	**59.1**	1.3	0.6486	0.7128	1.363	2.942	1.018	1.490
MEAR	267.4	239.0	28.4	**0.7710**	0.7796	1.385	3.572	1.140	2.050
ORGW	762.6	762.3	**0.3**	0.6572	**0.471**	1.601	**1.500**	**1.011**	2.000
ORGR	3217.0	3026.4	190.6	**0.4749**	0.5442	1.601	3.223	1.019	**1.050**
ORGDW	3060.7	3042.9	17.8	0.5379	0.5377	1.599	1.967	1.018	1.172
ORGDR	222.6	201.1	21.5	0.5782	0.5346	1.606	3.209	1.024	1.490
OTHW	279.4	269.1	10.3	0.6662	**0.8230**	1.339	**4.066**	1.047	1.652
OTHR	299.0	279.5	19.5	0.7143	0.7038	**1.326**	3.607	1.018	1.3
PERW	838.9	835.4	3.5	0.6567	0.6410	1.600	2.310	1.0356	1.436
PERR	**3863.0**	**3662.9**	**200.1**	0.6744	0.6675	1.517	3.004	1.073	1.431
PERDW	1678.7	1620.2	58.5	0.5696	0.5721	1.691	2.842	1.096	1.734
PERDR	1644.7	1543.2	101.5	0.5939	0.6101	**1.722**	3.404	**1.180**	**2.207**
TIMW	364.9	343.7	21.3	0.6512	0.5931	1.5356	2.555	1.048	1.580
TIMR	919.3	836.5	82.8	0.6142	0.7244	1.495	3.685	1.079	1.588
ALLW	1329.3	1311.0	18.3	0.6131	0.6244	1.527	2.641	1.037	1.443
ALLR	2409.0	2267.1	141.9	0.6234	0.6412	1.576	3.405	1.077	1.619

Table 7.3: Detailed information about the processed text fragments and the matched answer extraction patterns for non-NIL questions. The method used is the concatenation based one and the data is the training data.

for NIL questions are given in Table 7.4. The tables are given separately for the datasets consisting of non-NIL questions and of NIL questions because these two types of questions are quite different. If a NIL question is right, there were either no documents with a sufficiently high document similarity value or no AEPs matched. Thus, there is no information concerning these for the right answers. If a NIL question has received a wrong answer, there can be no right answers (i.e. the string *NIL*) in the list of answers returned by the system and thus no detailed information on the text fragments and AEPs for right answers. Thus, the table containing detailed information for NIL questions only gives information about questions that received a wrong answer and only about wrong answer candidates. From this table we can also observe that the number of text fragments is always equal to the number of (wrong) answer candidates. This is natural as there are no right answer candidates and as only one answer candidate is extracted from each text fragment. From Table 7.4 we can observe that only questions of type *Measure*, *Organization* and *Definition Organization* were answered incorrectly. When we compare the average figures (*ALLW*) with the corresponding figures for non-NIL questions (wrong answers for the *Type ALLW*), we notice that the average number of text fragments and answer candidates processed is much lower, 217.8 versus 1329.3 and 1311.0.

The average document similarity value is also much lower for NIL questions than for non-NIL questions; 0.4792 versus 0.6131. There is no important difference between NIL and non-NIL questions with regard to context size – 1.522 versus 1.527 and with regard to the number of QTags in the patterns – 1.018 versus 1.037.

Type	**Frag**	**ACand**	**DocSim**	**Context**	**QTags**
		Wrong	Wrong	Wrong	Wrong
MEAW	102.5	102.5	0.4472	1.371	1.027
ORGW	505	505	0.4613	1.733	1.012
ORGDW	161	161	0.5611	1.615	1.006
ALLW	217.8	217.8	0.4792	1.522	1.018

Table 7.4: Detailed information about the processed text fragments and the matched answer extraction patterns for NIL questions that are answered wrong. Concatenation based method and the training data.

7.2.2 Test data

This subsection presents the results of the experiments using the test data. The Baseline for the test data is 15/179 ≈ 8.4% which is obtained by answering *NIL* to every question. Table 7.5 presents the results for the whole dataset. The overall percentage of correct answers is quite a bit lower for test data (16.8 % and 26.3 %) than for training data (44.4 % and 50.6 %). Both the training and test data results show better performance on definition questions than on factoid questions. For NIL questions, the recall is the same for both datasets. However, precision and thus also F_1 measure are higher for the test data than for the training data.

run	Right % [179]	Wrong % [179]	X % [179]	AR Score [179]	Right % F [143]	Right % D [21]	NIL [15] P	NIL [15] R	NIL [15] F_1
concatTest$_1$	16.8	73.7	8.6	0.181	8.4	28.6	0.333	0.800	0.471
concatTest$_3$	26.3	63.7	9.5	0.181	18.2	42.9	0.333	0.800	0.471

Table 7.5: Results in the QA task using the concatenation based method on test data. The symbol X stands for inexact answers.

The class-specific results of the experiments are given in Table 7.6. When taking into account only the first answer, the best results (33.3%) are obtained for the class *Definition Organization* and the worst results (3.8%) for the class *Time*. When the three first answers are taken into account, the best results (55.6%) are obtained for the class *Definition Person* and the worst (5.9%) for the class *Measure*. In the non-NIL results for test data, taking into account three instead of just one answer improves the results much more than for the training data. For the test data, the improvement is 93.6% (from 11.0% to 21.3%) and for the training data, it is 17.3% (from 40.0% to 46.9%).

run	*Correct Answers %*								
	Definition		*Factoid*						*Both*
	ORG [12]	PER [9]	LOC [23]	MEA [17]	ORG [21]	OTH [32]	PER [24]	TIM [26]	[164]
concatTest$_1$	**33.3**	22.2	4.3	5.9	4.8	12.5	16.7	**3.8**	11.0
concatTest$_3$	33.3	**55.6**	21.7	**5.9**	14.3	12.5	33.3	18.9	21.3

Table 7.6: Percentage of correct answers for factoid and definition questions. The AEPs are formed using the concatenation based method and the experiments are performed using the test data.

Table 7.7 presents detailed information about the extracted text fragments and about the AEPs that extracted the answer candidates. The maximum average number of text fragments (7984.8) was extracted for questions of *Type ORGW* and the minimum average number of text fragments (17.0) was extracted for questions of *Type MEAR*. The maximum and minimum average numbers of wrong answer candidates were extracted for the same *Types*. The maximum average number of right answer candidates was extracted for questions of *Type ORGR* and the minimum average number of answer candidates for the *Type MEAW*. The greatest average document similarity value is 0.9925. It occurs for documents from which wrong answer candidates of *Type MEAR* have been extracted. The smallest document similarity value is 0.5314. It occurs for documents from which correct answer candidates for the *Type ORGDW* have been extracted. The difference in the average context size between wrong and right answer candidates for the *Types ALLW* and *ALLR* is much smaller for the AEPs that extracted answer candidates from the test data than for the AEPs that have extracted answer candidates from the training data. In the training data, the difference for the *Type ALLW* is 1.114 (The average context sizes are: 1.527 and 2.641.) and for the *Type ALLR* it is 1.829 (The average context sizes are: 1.576 and 3.405.) In the test data, the corresponding figure for the *Type ALLW* is 0.121 (The average context sizes are: 1.477 and 1.598.) and the figure for the *Type ALLW* is 0.563 (The average context sizes are: 1.485 and 2.048.). The table also shows a similar phenomenon as the one occurring for the size of the context for the number of QTags. The average number of QTags in the AEPs is also smaller for the right answers of the test data than for the right answers of the training data – 1.169 versus 1.1443 for *ALLW* and 1.382 versus 1.619 for *ALLR*. Thus, also the difference between the number of QTags for right and wrong answers is smaller for the test data than for the training data.

Table 7.8 presents detailed information on the processed text fragments and on the AEPs that matched the answer candidates. The only question classes for which wrong NIL answers were produced are *Location*, *Organization* and *Person*.

Type	Frag	ACand		DocSim		Context		QTags	
		Wrong	Right	Wrong	Right	Wrong	Right	Wrong	Right
LOCW	2031.2	2014.9	16.4	0.5688	0.5958	1.585	1.601	1.029	1.150
LOCR	3734.7	3639.7	95.0	0.7884	0.8547	1.463	1.652	1.057	1.286
MEAW	191.8	190.8	**1.1**	0.5751	0.7288	1.365	1.375	1.052	**1.000**
MEAR	**17.0**	**11.0**	6.0	**0.9925**	**0.9793**	1.364	1.500	**1.273**	1.500
ORGW	**7984.8**	**7969.6**	15.3	0.5802	0.6043	1.556	1.636	1.021	1.049
ORGR	3089.0	2913.0	**176.0**	**0.5317**	0.6128	**1.765**	**3.472**	**1.008**	1.000
ORGDW	2280.6	2278.3	2.4	0.5851	**0.5314**	1.553	2.158	1.014	1.316
ORGDR	939.3	808.7	130.7	0.6165	0.6187	1.525	2.698	1.022	1.448
OTHW	701.6	696.8	4.8	0.6327	0.5815	1.322	1.301	1.038	1.109
OTHR	659.3	623.7	35.7	0.6641	0.6905	**1.264**	**1.235**	1.026	1.146
PERW	1617.0	1599.0	18.4	0.5628	0.5504	1.517	1.453	1.045	1.174
PERR	564.5	539.0	25.5	0.6720	0.7677	1.598	1.443	1.018	1.057
PERDW	5464.4	5343.6	120.9	0.6065	0.6716	1.611	2.178	1.053	2.038
PERDR	2554.5	2525.0	29.5	0.6622	0.8038	1.623	2.568	1.031	**2.406**
TIMW	503.6	487.4	16.2	0.6235	0.6248	1.477	1.7499	1.090	1.083
TIMR	999.0	936.0	63.0	0.5575	0.5752	1.433	3.0159	1.017	1.048
ALLW	2220.2	2203.5	16.6	0.5948	0.6058	1.477	1.598	1.047	1.169
ALLR	1646.4	1575.3	71.2	0.6848	0.7376	1.485	2.048	1.044	1.382

Table 7.7: Detailed information about the processed text fragments and the matched AEPs for the concatenation based method run on the test data set.

Type	Frag	ACand	DocSim	Context	QTags
		Wrong	Wrong	Wrong	Wrong
LOCW	58.0	58.0	0.5717	1.448	1.000
ORGW	8.0	8.0	0.623	1.000	1.000
PERW	37.0	37.0	0.526	1.378	1.027
ALLW	34.3	34.3	0.5735	1.276	1.009

Table 7.8: Detailed information about the processed text fragments and the matched answer extraction patterns for NIL questions that are answered wrong. The method for producing the AEPs is the concatenation based one and the data is the test data.

7.3 Alignment based method

This section presents the results of the experiments with the alignment based method. The results with training data are given first and the results with the test data are given after them.

7.3.1 Training data

Table 7.9 shows the results of the alignment based method on training data. The percentage of right answers is 47.2 when only the first answer is considered. This is a bit better than 44.4% obtained with the concatenation based method on the same data.

run	Right % [180]	Wrong % [180]	X % [180]	AR Score [180]	Right % F [115]	Right % D [45]	NIL [20] P	NIL [20] R	NIL [20] F_1
alignTrain$_1$	47.2	47.8	5.0	0.496	37.4	57.8	0.276	0.800	0.410
alignTrain$_3$	51.1	44.4	4.4	0.496	40.9	64.4	0.276	0.800	0.410

Table 7.9: Results for the alignment based method on training data. The symbol X stands for inexact answers.

Table 7.10 shows the class-specific results. The results are best for the class *Definition Person* and worst for the class *Other*. This is also the case for the results of the concatenation based method with the same data.

run	*Correct Answers %*								
	Definition		*Factoid*						*Both*
	ORG [21]	PER [24]	LOC [21]	MEA [20]	ORG [14]	OTH [15]	PER [29]	TIM [16]	[160]
alignTrain$_1$	47.6	**66.7**	42.9	40.0	50.0	**20.0**	34.5	37.5	43.1
alignTrain$_3$	47.6	**79.2**	42.9	45.0	50.0	**20.0**	41.4	43.8	47.5

Table 7.10: Percentage of correct answers for factoid and definition questions for the alignment based method and training data. The breakdown of the results is done according to answer type.

Table 7.11 shows detailed information on the text fragments and on the AEPs. The highest average numbers of text fragments (4782.2) as well as of wrong (4529.5) and right (252.8) answer candidates occur for the question class that also has the highest number of correct answers, i.e. the class *Definition Person*. The smallest average numbers of text fragments and of wrong and right answer candidates occur for the class that also has the lowest number of correct answers, i.e. the class *Factoid Other*. The highest

average document similarity value – 0.8218 – occurs for the *Type OTHW*. The lowest average document similarity value – 0.5137 – occurs for the *Type ORGW*. The greatest average context size is 5.000 and it occurs for *OTHW*. The smallest average context size is 1.408 and it occurs for *MEAW*. The highest average number of *QTags* occurs for *OTHW* (2.333) and the lowest for *ORGW* (1.013.)

Type	**Frag**	**ACand**		**DocSim**		**Context**		**QTags**	
		Wrong	Right	Wrong	Right	Wrong	Right	Wrong	Right
LOCW	1970.5	1946.1	24.5	0.5798	0.6457	1.647	**1.971**	1.033	1.320
LOCR	1207.4	1101.1	106.3	0.5679	0.6142	1.697	3.528	1.032	1.325
MEAW	145.6	134.9	10.7	0.6368	0.6607	**1.408**	3.144	1.020	1.420
MEAR	338.0	307.5	30.5	**0.7164**	0.7351	1.449	3.696	1.108	1.860
ORGW	1140.3	1133.8	6.5	0.6237	**0.5137**	1.742	2.680	**1.013**	1.650
ORGR	2803.2	2637.5	165.7	0.6499	0.6961	1.650	3.497	1.027	**1.267**
ORGDW	2091.2	2083.9	7.3	0.5735	0.5892	1.606	2.230	1.015	1.414
ORGDR	654.3	618.2	36.1	**0.5650**	0.5374	1.634	3.350	1.022	1.361
OTHW	56.7	53.5	**3.2**	0.6762	**0.8218**	1.6305	**5.000**	1.723	**2.333**
OTHR	**29.0**	**4.0**	25.0	0.6802	0.7072	**2.500**	4.000	**2.000**	1.600
PERW	715.0	710.9	4.1	0.6489	0.6203	1.529	2.168	1.036	1.492
PERR	3451.0	3261.9	189.1	0.6738	0.6826	1.478	3.454	1.061	1.365
PERDW	1484.8	1427.9	56.9	0.6000	0.6388	1.608	3.315	1.092	1.904
PERDR	**4782.2**	**4529.5**	**252.8**	0.5832	0.6052	1.702	3.155	1.127	1.998
TIMW	255.5	236.5	19.0	0.6704	0.6247	1.713	2.719	1.040	1.828
TIMR	742.2	667.2	75.0	0.6002	0.6886	1.610	3.732	1.078	1.598
ALLW	931.0	915.9	15.1	0.6259	0.6372	1.592	2.793	1.127	1.650
ALLR	2284.1	2149.3	134.8	0.6157	0.6396	1.632	3.443	1.085	1.573

Table 7.11: Detailed information about the processed text fragments and the matched answer extraction patterns. The method is the alignment based and the data is the training data.

Type	**Frag**	**ACand**	**DocSim**	**Context**	**QTags**
		Wrong	Wrong	Wrong	Wrong
MEAW	136.0	136.0	0.5525	1.684	1.000
ORGW	253.0	253.0	0.4503	1.664	1.017
OTHW	40.0	40.0	0.4540	1.250	1.050
ALLW	170.5	170.5	0.4768	1.565	1.021

Table 7.12: Detailed information about the processed text fragments and the matched AEPs for NIL questions that are answered incorrectly. The method is the alignment based and the data is the training data.

Table 7.12 presents detailed information about the processed text fragments and the AEPs that matched the answer candidates for NIL questions. The results for the alignment based method are quite similar to those for

the concatenation based method tested with the same data. The question classes for which a wrong answer was provided are the same except that instead of producing an non-NIL answer for the class *Definition Organization*, the alignment based method produced a non-NIL answer for the class *Other*.

7.3.2 Test data

Table 7.13 shows the results of the experiments using the test data. The proportion of right answers is quite the same as in the concatenation based method. Results that take into account only the first answer candidate are slightly better for the alignment based method than for the concatenation based method; 17.3% versus 16.8%. However, when the three best answers are taken into account, the results of the concatenation based method are slightly better; 26.3% versus 25.1%.

run	**Right** % [179]	**Wrong** % [179]	**X** % [179]	**AR** **Score** [179]	**Right** % F [143]	% D [21]	**NIL** [15] P	R	F_1
alignTest$_1$	17.3	73.2	9.5	0.221	7.7	33.3	0.317	0.867	0.464
alignTest$_3$	25.1	64.8	10.1	0.221	17.5	33.3	0.317	0.867	0.464

Table 7.13: Results for the alignment based method on test data. The symbol X stands for enexact answers.

Table 7.14 presents the results for each question class and only for faction questions. From this table we can clearly notice that the better results of the alignment based method are solely due to better performance for NIL questions. The percentage of correct answers for factoid questions – 11% – is excatly the same as in the concatenation based method when only the first answer is taken into account. The percentage of correct answers is lower for the alignment based method than for the concatenation based method when the three first answers are taken into account: 19.5% versus 21.3%.

run	*Correct Answers %*								
	Definition		*Factoid*						*Both*
	ORG [12]	**PER** [9]	**LOC** [23]	**MEA** [17]	**ORG** [21]	**OTH** [32]	**PER** [24]	**TIM** [26]	[164]
alignTest$_1$	**33.3**	**33.3**	4.3	11.8	4.8	6.3	16.7	**3.8**	11.0
alignTest$_3$	**33.3**	**33.3**	17.4	11.8	19.0	**9.4**	25.0	23.1	19.5

Table 7.14: Percentage of correct answers for factoid and definition questions for the alignment based method and test data. The results are presented according to answer type.

Table 7.15 presents detailed results on the text fragments and on the AEPs. From the table we may observe that the average of the *Type*-specific averages (i.e. the figures for *Types ALLW* and *ALLR*) for document similarity, context size of the AEPs and the number of QTags is greater for

right answers than for wrong answers. This does not necessarily hold for *Type*-specific figures. For example, the document similarity value for the *Type ORGDW* is greater for wrong answer candidates than for right answer candidates.

Type	**Frag**	**ACand**		**DocSim**		**Context**		**QTags**	
		Wrong	Right	Wrong	Right	Wrong	Right	Wrong	Right
LOCW	1718.5	1710.5	8.0	0.5791	0.5774	1.694	1.689	1.032	1.213
LOCR	53.0	47.0	6.0	**0.9122**	**1.0000**	1.563	1.611	1.076	1.278
MEAW	152.4	152.1	0.3	0.5718	0.7628	1.401	**1.417**	1.018	**1.000**
MEAR	119.5	113.5	6.0	0.8243	0.8751	1.387	1.750	1.153	1.125
ORGW	**5572.3**	**5529.4**	42.9	0.5796	0.6243	1.686	1.688	1.023	1.073
ORGR	2336.0	2181.0	**155.0**	**0.5311**	0.6065	**1.906**	**3.645**	1.010	**1.000**
ORGDW	1125.1	1123.3	1.9	0.6231	0.5381	1.617	2.067	**1.008**	1.200
ORGDR	763.0	668.0	95.0	0.6142	0.5955	1.586	2.761	1.019	1.384
OTHW	**8.2**	**8.1**	**0.2**	0.5884	0.5610	1.323	**1.417**	1.874	2.000
OTHR	53.0	52.0	1.0	0.6726	**0.5000**	**1.269**	2.000	**1.904**	2.000
PERW	918.5	907.4	11.1	0.5822	0.5425	1.474	1.424	1.040	1.234
PERR	740.0	708.0	32.0	0.7632	0.7327	1.520	1.750	1.028	1.125
PERDW	646.5	639.2	7.3	0.7612	0.8132	1.616	1.990	1.054	1.781
PERDR	606.3	566.3	40.0	0.7820	0.8559	1.615	2.756	1.108	**2.309**
TIMW	394.5	384.9	9.6	0.6298	0.6469	1.573	1.898	1.071	1.115
TIMR	842.0	788.0	54.0	0.5572	0.5639	1.528	3.296	1.013	1.056
ALLW	1294.0	1283.4	10.6	0.6018	0.6224	1.536	1.710	1.169	1.235
ALLR	601.7	553.8	47.9	0.7276	0.7505	1.552	2.426	1.128	1.505

Table 7.15: Detailed information about the processed text fragments and the matched AEPs. The method used is the alignment based and the data is the test data.

Table 7.16 presents detailed information about the processed text fragments and the AEPs for NIL questions. The questions that were answered wrongly belong to only two classes: *Other* and *Time*. All NIL questions in both of these question categories were answered correctly using the concatenation based method with the same data.

Type	**Frag**	**ACand**	**DocSim**	**Context**	**QTags**
		Wrong	Wrong	Wrong	Wrong
OTHW	50.0	50.0	0.5836	1.520	1.000
TIMW	39	39	0.5261	1.256	1.026
ALLW	44.5	44.5	0.5548	1.388	1.013

Table 7.16: Detailed information about the processed text fragments and the matched AEPs for NIL questions that are answered incorrectly. The method is the alignment based and the data is the test data.

7.4 Comparison of all results

The two figures of this section (Figures 7.1 and 7.2) present a summary and comparison of the results presented in the two previous sections. Figure 7.1 presents the information of the four tables that show experimental results for the whole datasets(i.e. Tables 7.1, 7.5, 7.9 and 7.13) in a single picture so that the different runs can easily be compared with each other. The x-axis shows the names of the columns present in the tables. The names *Right*, *Wrong*, *Inexact* and *AR Score* are exactly the same. The names *RightF* and *RightD* mean right factoid questions and right definition questions, respectively. The name *NILP* stands for NIL precision, *NILR* for NIL recall and *NILF1* for NIL F_1-measure.

The y-axis illustrates the amount of difference of a run with the baseline run, i.e. $concatTrain_1$, which is marked by a horizontal line at 0. The difference of a run with regard the baseline run has been calculated simply by subtracting the figure for the baseline from the corresponding figure of the run in question. For example, the percentage of right answers for the baseline is 44.4% and for the run $concatTrain_3$ it is 50.6%. The difference between the runs is 6.2, as can be seen in the figure. In the case of the *AR Score*, *NILP*, *NILR* and *NILF1* where the figures range from 0 to 1 and not from 0 to 100 as is the case for the other figures, the difference has simply been multiplied by 100 in order to obtain figures of the same magnitude. The numbers 1 and 3 have been omitted from the labels of the runs for the columns *AR Score*, *NILP*, *NILR* and *NILF1* because the results of the runs are the same irrespective of whether they only take the first answer or the best of the first three answers. If two runs have produced same results in the other columns, their names have been merged in a self-describing way, such as $concat\&alignTest_1$ in column *Inexact*, which means that the runs $concatTest_1$ and $alignTest_3$ have produces the same percentage of inexact answers.

The run $concatTrain_1$ was chosen for baseline because the concatenation based method is a simpler one than the alignment based one and because choosing just the first answer instead of the three first ones is the basic case in the CLEF evaluation campaigns of the years 2004 and 2005 from which the data comes. Running the method on training data should produce quite good results as the task is considerably more easy than when it is run on test data.

The columns *Right*, *Wrong*, *RightF* and *RightD* in figure 7.1 show that in general, the runs performing better than the baseline are $alignTrain_3$, $concatTrain_3$ and $alignTrain_1$, in the order of citation. The column showing

the difference of wrong answers has to be read in the opposite direction than the three other columns, i.e. the greater the negative difference is, the better the results of the run are in comparison with the baseline run. The column showing the results of the *AR Score* show that when all answers are taken into account, the runs using the alignment based method perform clearly better than the runs using the concatenation based method. The columns *NILP*, *NILR* and *NILF1* present results that are very different from the results that take into account all questions. The best results are obtained on test data and not on training data which is unexpected. However, the results of the different runs are very similar with each other, which again is very different from the results presented in the other columns.

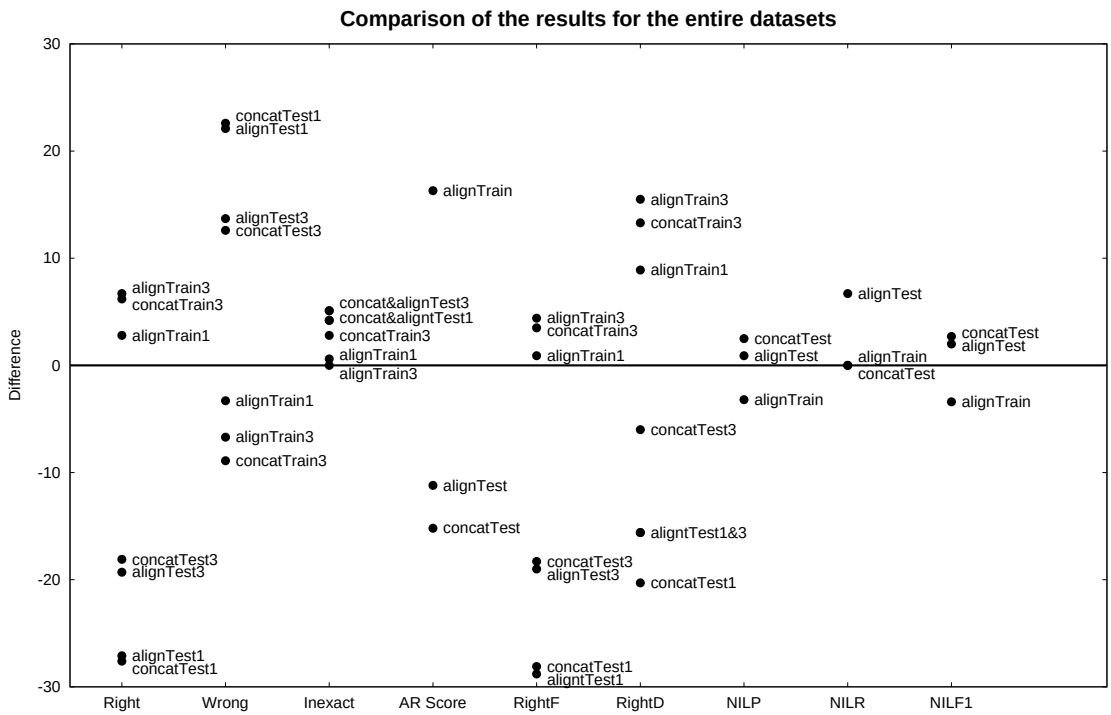


Figure 7.1: Comparison of results presented in Tables 7.1, 7.5, 7.9 and 7.13. The baseline, which is the run $concatTrain_1$, is marked by the straight line at 0.

Figure 7.2 presents the results of those tables that contain class-specific results for non-NIL questions, i.e. Tables 7.2, 7.6, 7.10 and 7.14, in a single picture. The x-axis represents the different question classes and the y-axis represents the difference of runs in points of percentage from the baseline.

As in Figure 7.1, the baseline is the run *concatTrain*$_1$. In the cases where several runs have the same result, either the complete names of the runs are marked or the names of the runs are merged in a self-describing manner. The decision of which notation to use is based solely on the readability of the picture and does not bear any other meaning.

The class-specific analysis of the results of the non-NIL questions presented in Figure 7.2 is in line with the overall results given in Figure 7.1 - except for the results concerning only NIL questions. As the column *ALL* shows, the best three runs are the same as in Figure 7.1. The best results in comparison with the baseline are obtained by the run *alignTrain*$_3$ in the class *PERD*. The worst results in comparison with the baseline are obtained by the runs *concatTest*$_1$ and *concatAlign*$_1$ in the class *ORG*. However, the runs *concatTest*$_1$ and *concatTest*$_3$ in class *MEA* perform nearly as poorly in comparison with the baseline.

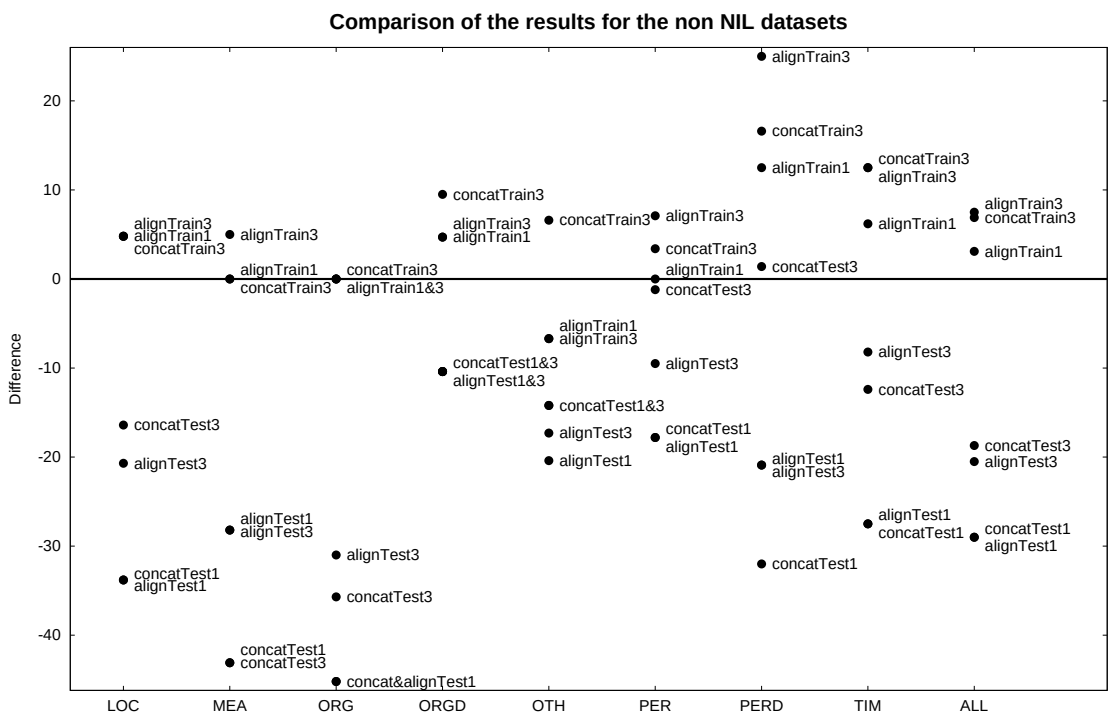


Figure 7.2: Comparison of results presented in the Tables 7.2, 7.6, 7.10 and 7.14. The baseline, which is the run *concatTrain*$_1$, is marked by the straight line at 0.

7.4.1 New answers found

The five figures of this section list completely new answers that the concatenation (C) and alignment (A) based methods found in the data. Figures 7.3 and 7.4 present new answers found in the training data and the Figures 7.5, 7.6 and 7.7 list new answers found in the test data. As can be seen in the figures, the new answers are judged either right (R) or inexact (X). The judgment is done by the author of this thesis and is based on common knowledge of the world. Wrong new answers are naturally not listed at all because they were numerous and useless.

The figures show that there is no great difference between the number and quality of novel right answers found by the concatenation and alignment based methods. However, the concatenation based method found 7 inexact answers in the training data and the alignment based method as many as 30. This same phenomenon is not present in the test data where the concatenation based method found 16 inexact answers and the alignment based one 14. As many as 20 of the inexact answers found by the alignment based method in the training data belong to only two questions. It may be argued that the great number of inexact answers is only a question specific phenomenon. The number of new answers found is 26 (concatenation based method) and 28 (alignment based method) for the training data and 49 (concatenation based method) and 48 (alignment based method) for the test data. The greater amount of new answers found in the test data can be explained by the fact that the AEPs of the methods were trained on the training data and thus more easily find the answers present in the training data than in the test data. However, one would expect that the alignment based methods would find more new answers than the concatenation based methods both in the test data and in the training data because the AR Score is higher for the alignment based runs.

Who was Yasser Arafat? PLO Chairman			
Palestinian Liberation Organization Chairman	R	C	A
PLO Chairman	R	C	A
PLO chief	R	C	A
PLO head	R	C	A
PLO leader	R	C	A
Who is Giulio Andreotti? former Italian Prime Minister			
Former Italian premier	R	C	
Name a building wrapped by Christo. Reichstag			
Berlin's reichstag	R	C	A
Where is the ozone hole? Antarctic			
Antarctica	R	C	A
Name a German car producer. Volkswagen			
Mercedez-Benz	R	C	
BMW	R	C	A
Porsche	R	C	
Trabant	X	C	
Name a film in which computer animation was used. The Pagemaster			
The Little Mermaid	R	C	
The Lion King	R	C	
Jurassic Park	R	C	
Who is Goodwill Zwelithini? Zulu King			
Zulu monarch	R	C	A
Who was Haiti's former military commander in chief? Lt. Gen. Raoul Cedras			
Raoul Cedras	R	C	A
Who is El Nino named after? the Christ child			
Christ	X	C	
Who was the President of the United States of America between 1976 and 1980? Carter			
Jimmy Carter	R	C	A
Who praised Indurain for his "capacity for sacrifice, class and healthy spirit of competition", when he won the Tour de France for the fourth time? Spanish Prime Minister Felipe Gonzalez			
Prime Minister Felipe Gonzalez	X	C	A
Which two scientists discovered "G proteins"? Alfred G. Gilman, 53, of the University of Texas Southwestern Medical Center in Dallas and Martin Rodbell			
Rodbell	X		A
Gilman	X		A
Who was Charles Bukowski? writer and poet laureate			
Famous poet	X	C	A
Poet	X		A
Los Angeles writer	R	C	A
Poet and fiction writer	R	C	A
Writer	X	C	A
Prolific writer	X		A
Laureate of Los Angeles	X		A
Noted author	X		A
Los Angeles poet	R		A

Figure 7.3: The new answers found by the concatenation (C) and alignment (A) based methods to the questions in the training data. The question and its answer in the training data are given in bold above each new answer. The new answers are judged either R (right) or X (ineXact). Wrong new answers are not listed. The list of answers is continued in Figure 7.4

.

Who is Willy Claes? Nato secretary-general			
Nato secretary general	R		A
Former Belgian foreign minister	R		A
Secretary-general of Nato	R		A
Former Belgian minister	X		A
Economics affairs minister	X		A
Belgian Flemish Socialist	X		A
Belgiums's economics minister	X		A
Belgian socialist	X		A
Belgian government minister	X		A
Former foreign minister	X		A
Nato officials	X		A
Belgium's financeminister	X		A
European politicians	X		A
Flemish socialist	X		A
Who is Rolf Ekeus? UN weapons envoy			
The UN's chief arms inspector	R	C	A
Who is Dzhokhar Dudayev? Chechen leader			
Chechen rebel leader	R	C	A
Chechen separatist leader	R	C	
Who is Yasushi Akashi? UN special envoy			
UN peacekeeping chief	R	C	A
What is the EZLN? Zapatista Army of National Liberation			
Zapatista National Liberation Army	R	C	A
Who is Joao Havelange? FIFA's Brazilian president			
FIFA president	R	C	A
Who is Javier Solana? the Spanish Foreign Minister			
Foreign Minister	X	C	
Who is Antonio Di Pietro? Milan Magistrate			
Magistrate	X	C	A
Former Corruption-busting magistrate	R		A
Former star magistrate	X		A
Chief prosecutor of Milan	R		A
Investigating magistrate	X		A
Milanese investigating magistrate	R		A
Fraud buster	X		A
Corruption-busting magistrate	X		A
Lawyer	X		A
Respected figure	X		A
Consultant	X		A
Anti-corruption magistrate	X		A
What is the Natural Resources Defense Council? an activist environmental group			
Wilderness alliance	X		A
Wilderness society	X		A
What is the WEU? the embryo defence arm of the EU			
Western european union	R	C	A
What is the PRI? the Institutional Revolutionary Party			
Partido revolucionario institucional	R		A
Partido nacional revolucionario	R		A
Which city is the Al Aqsa Mosque in? Jerusalem			
East Jerusalem	X		A

Number of questions for which a new right (R) answer was found: 16 (C), 17 (A)
Number of new right (R) answers: 26 (C), 28 (A)
Number of inexact (X) answers: 7 (C), 30 (A)

Figure 7.4: The new answers found by the concatenation (C) and alignment (A) based methods to the questions in the training data. Continuation of Figure 7.3

What is the Antarctic continent covered with? ice-cap			
Ice	R	C	
What is the ozone hole caused by? man-made chlorine			
Chlorofluorcarbons	R	C	A
Chlorine	X	C	A
Name a pesticide. DDT			
Malathion	R	C	A
Dioxins	R	C	
Alachlor	R	C	
Benlate	R	C	
Insecticide	X	C	
Fungicide	X	C	
Herbicide	X	C	
Sulfur	R	C	
Chlordane	R	C	
Carbendazim	R	C	
Soap	R	C	
Organo-chlorine	R		A
Who is Jean-Bertrand Aristide? Haiti's first democratically elected president			
President	X	C	A
President of Haiti	R	C	
Haitian president	R	C	A
Haiti's elected president	R	C	
Exiled Haitian resident	R	C	A
Haitis's president	R	C	
Haiti's exiled president	R	C	
Who was the embargo against Iraq imposed by? The U.N. Security Council			
United Nations	R	C	A
Name a cetacean. whale			
Dolphin	R	C	A
How many people speak Gaelic in Scotland? 1.4 percent of Scots			
60,000	R	C	A
Where is the Al Aqsa Mosque? Jerusalem			
East Jerusalem	R	C	A
Middle East	X		A
Israel	X		A
Holy Land	X		A
Where is the Valley of the Kings? in Thebes			
Egypt	R	C	A
Nile	X	C	
When did Prince Charles and Diana get married? 1981			
In the early 1980s	R	C	A
What is UEFA? European football's governing body			
European players' union	X	C	A
Football association	X	C	A
Footballers' association	X	C	A
What year was Halley's comet visible? 1909			
1986	R	C	A
Who is Paul Simon? politician			
American singer	R	C	A
Singer-songwriter	R	C	A
Singer	R	C	A

Figure 7.5: The new answers found by the concatenation (C) and alignment (A) based methods to the questions in the test data. The question and its answer in the test data are given in bold above each new answer. The new answers are judged either R (right) or X (ineXact). Wrong new answers are not listed. The list of answers is continued in Figures 7.6 and 7.7.

Name a board game. Scrabble			
Temple	R	C	
Cluedo	R	C	
Monopoly	R	C	A
Who is Silvio Berlusconi? a conservative media magnate			
Prime Minister	X	C	A
Italy's former Prime Minister	R	C	
Former Italian Prime Minister	R	C	A
Prime Minister of Italy	R	C	
Italy's Prime Minister	R	C	
Thin-skinned Italian Prime Minister	X	C	
Italian Prime Minister	R	C	A
Sitting Italian Prime Minister	R	C	
What did the artist Christo wrap up? a Paris bridge			
Reichstag	R	C	A
What does a luthier make? guitar			
instrument	X	C	
What is freemasonry? a secret society			
brotherhood	X	C	A
Who is Alan Turing? the British scientist whom the authors rediscover as the uncrowned king of modern artifical intelligence			
Mathematician	R	C	A
Name a country that exports rice. Vietnam			
Middle East and Brazil	X	C	A
Thailand	R	C	A
The United States	R	C	A
Burma	R	C	A
Brazil	R	C	A
Name a fast food chain. Burger King			
KFC	R	C	A
Little Chef	R	C	A
Happy Eater	R	C	A
McDonald's	R		A
What is the UNHCR? UN High Commissioner for Refugees			
United Nations High Commissioner for Refugees	R	C	
Who is Juan Antonio Samaranch? IOC president			
International Olympic Committee	X	C	A
Olympic President	X	C	A
Olympic Chief	X		A
Where is Hyde Park? N.Y.			
London	R	C	A
What is the world's highest mountain? Everest			
Mount Everest	R	C	A
What does Oracle sell? software and systems services for British Telecommunications' planned delivery of interactive multimedia services to homes			
Database	R	C	A
Software	R	C	A
Where is Halifax located? Canada			
England	R	C	A
Who is the new president of Rwanda? Pasteur Bizimungu			
State Pasteur Bizimungu	R	C	A

Figure 7.6: The new answers found by the concatenation and alignment based methods applied on test data. Continuation of Figure 7.5. The list of new answers is continued in Figure 7.7

Name an oil company. Conoco		
Monica-based Macpherson	R	A
Alpetrol	R	A
Hydro-Congo	R	A
Cleveland-based Lubrizol	R	A
Pennzoil	R	A
Lukoil	R	A
Petrovietnam	R	A
Shell	R	A
ARCO	R	A
Agip	R	A
Norsk Hydro	R	A
Komineft	R	A
Mobil	R	A
Lubrizol	R	A

Number of questions for which a new right (R) answer was found: 22 (C), 22 (A)
Number of new right (R) answers: 49 (C), 48 (A)
Number of inexact (X) answers: 16 (C), 19 (A)

Figure 7.7: The new answers found by the concatenation and alignment based methods applied on test data. Continuation of Figures 7.5 and 7.6.

Chapter 8

Discussion

In this chapter, the nature and significance of the results of the experiments presented in the previous chapter are analyzed. The first section begins with a general discussion on the experimental setting and the results and continues with a detailed analysis of the results given in the tables and figures of the previous chapter. The second section discusses the problems in categorizing the answers into the classes right, inexact and wrong. The third section presents a comparison between the novel answer extraction methods and the existing ones. Finally, the fourth subsection describes some limitations of the present work and gives ideas for future work.

8.1 Analysis of the experimental results

The experiments on the novel answer extraction methods have been conducted by incorporating them into a QA system. This was done in order to make the results more easily comparable with the results of entire QA systems and because a standard IR method was readily available. However, the incorporation of the methods into a QA system does affect the results and it has to be taken into account in their analysis. The question classification part works perfectly, as the correct question class is given as input to the system. The extraction of the query terms from the question is performed using a very simple method which is explained in Section 5.1. The IR method used is also a simplistic one, and it is also described in Section 5.1. The reason for using very simple methods in these components of the system is that their development is out of the scope of the present work. However, it would be an interesting and important direction for future work. One could suspect that the naive implementations of the two components would rule out the gain in performance obtained

by the perfectly working question classifier, but this is only speculation as no study has been carried out on the effect of the different components of our experimental setting on the overall performance. One way to eliminate all impact of the other components of the QA system when evaluating the performance of the answer extraction methods is to assume that the previous components in the pipeline work perfectly and to provide the novel methods a perfect input. This would raise the question of what a perfect input to an answer extraction method would look like. It could be a single paragraph containing the answer to the question and as many question words as possible. This would be an unrealistic experimental setting as developing perfect components for query word formation and the IR part is not feasible. As far as we know, no answer extraction method has been tested in isolation, but always as a part of a QA system.

The results of the QA system, 17.3% of right answers with the alignment based method on test data and 16.8% with the concatenation based method, are a bit below 23.7%, which is the average of the performances of the QA systems that participated in the CLEF 2004 QA evaluation [MVA+05]. However, as we have seen in Section 3.1, the answer extraction parts of most of the systems that have performed well in the CLEF QA systems evaluation campaign require a significant amount of hand-crafted AEPs, while the new methods presented in this thesis require no manual work at all.

When comparing the performances of the two novel methods with each other, one has to note that the overall figures on the test data are surprisingly similar. Before performing the experiments, the hypothesis was that the alignment based method would under-perform the concatenation based method on training data, but that it would outperform the concatenation based method on test data. This was because it was expected that the process of alignment would perform generalizations and thus the patterns would be less fitted to the training data, which would result in worse performance. However, the generalization was supposed to be useful on test data, but the differences are not very important. On training data, the alignment based method finds 43.1% of the factoid and definition answers and the concatenation based method finds 40.0%. On the test data, both methods find 11.0% of the answers to factoid and definition questions. Here we use the figures for factoid and definition questions only instead of the overall figures also containing the NIL questions, as the impact of the document retrieval module on the performance of the datasets containing NIL questions seems more important than on the datasets not containing NIL questions. This can be inferred from Table 4.4 on page 37 which shows that the proportion

of NIL questions answered correctly (i.e. where the query corresponds to 0 documents) even without any answer extraction component is 11/20 for training data and 8/15 for the test data. In other words, the number of right answers without an answer extraction component would be 11/160 (6.9%) for the training data and 8/164 (4.9%) for the test data.

As can be expected, both methods benefit considerably from taking into account the three first answers instead of just the first answer when tested using test data. The increase in performance for the concatenation based method is from 11.0% to 21.3%, or 11.3 points as can be observed in Table 7.6 (on page 84) and Figure 7.2 (on page 94). For the alignment based method the increase in performance is from 11.0% to 19.5%, or 8.5 points as can be observed in Table 7.14 (on page 90) and Figure 7.2 (on page 94). The difference when testing using training data is not nearly as important: 6.9 points for the concatenation based method and 4.4 points for the alignment based method. The reason for not benefiting from taking into account more answers as much for training data as for test data might be that the AEPs of both methods are over-fitted to the training data. This would be very natural as the AEPs have been derived from that data.

The differences in the training and test data affect the results. According to the question difficulty metric introduced in Section 4.3, the questions of the test data set are more difficult than the questions of the training data set. In addition, as presented in Tables 4.2 (on page 35) and 4.3 (on page 35), the answers are quite different in the datasets: the test data set contains more MUC-7 type entities and expressions than the training data set. Especially the answers for the question class *Person* differ a lot in the training and test data. In the training data, only 62.1% of the answers for questions belonging to the class *Person* are named entities of type *Person* according to the MUC-7 classification, whereas in the test data, all answers of the question class *Person* are named entities of the type *Person* according to the MUC-7 classification. This difference would suggest a poor answer extraction result for questions of class *Person* for the test data because the patterns have been formed from a quite dissimilar data. However, the experiments show that the drop in performance between training and test data in the question class *Person* is smaller than the drop in average. This can be observed in Figure 7.2 by looking at the differences in performance for the class *Person* and for all of the classes (denoted by *ALL*).

The tables containing detailed information about the processed text fragments and matched AEPs (i.e. Tables 7.3, 7.7, 7.11, 7.15, 7.4, 7.8, 7.12 and 7.16) show interesting facts. The number of text fragments processed or the number of right answers among the answer candidates does not seem

to affect the performance of the answer extraction methods. This can be seen in the tables which indicate that for both methods, the average number of processed text fragments is considerably higher for right answers than for wrong answers in the experiments performed on training data, but for the experiments performed on test data, the situation is the opposite. Wrong answers to NIL questions are all extracted from a relatively small number of text fragments. The average number of right answer candidates is always considerably smaller than the number of wrong answer candidates. The results do not differ when observing the answer candidates for questions for which eventually a right answer was chosen and when observing the answer candidates for questions for which a wrong answer was chosen. For NIL questions, the average number of wrong answer candidates is always quite small when compared to the average number of wrong answer candidates for non-NIL questions.

We have just observed that the numbers of processed text fragments or of right answer candidates extracted do not affect the results of the answer extraction methods. However, from the same tables we can observe that high document similarity values, large sizes of pattern context and a large number of question words in the AEP do correlate with the correctness of the extracted answer candidate. Here the number of question words in the pattern means simply the number of QTags and not the maximum QTag by which the AEPs have been classified when applying the method. These fields correspond to three of the terms appearing in Equation 6.2 on page 75, which is used to score the answers when performing answer selection . The field *DocSim* corresponds to $DocSim_{AC}$, *Context* corresponds to $Context_{AC}$ and *QTags* corresponds to $QTags_{AC}$. In addition, the equation contains the terms $Length_{AC}$ and $Freq_{AC}$, which take into account the length and frequency of an answer candidate.

The hypothesis for document similarity was that for questions that are answered wrong the document similarity for wrong answer candidates would be higher than for right answer candidates. This would explain why the wrong answer was chosen. For questions that have received correct answers, the document similarity for right answer candidates should be higher than that for the wrong answer candidates. However, by studying the document similarity values of the answer candidates (see the following tables in the previous chapter: Table 7.3 on page 82, Table 7.7 on page 86, Table 7.11 on page 7.11 and Table 7.15 on page 7.15), we can observe that this hypothesis does not hold. The right answer candidates always have higher document similarity values than the wrong answer candidates, irrespective of whether the answer candidate chosen was right or wrong. The

reason for this seems to be twofold: firstly, in the formula used for scoring the answers (Equation 6.2 on page 75), document similarity accounts only for 1/5 of the score and secondly, the document similarity scores are quite close to each other, the minimun being 0.4 and the maximum being 1.

The hypothesis and results for the values of the fields *Context* and *QTags* presented in the tables containing detailed information on the AEPs are similar to the hypothesis and results for document similarity. If the hypothesis did hold, then making the values less important in the answer scoring function (i.e. Equation 6.2 on page 75) might make the results better. On the other hand, if the opposite was true, i.e. the values for right answer candidates were systematically better than for wrong answer candidates irrespective of the correctness of the answer provided by the system, then the importance of these values in the scoring equation should be increased. Now the conclusion that can be drawn from the figures is that increasing the importance of the values for document similarity, context size and QTags might improve the results, but that further investigation is needed. The effect of the document similarity and the number of QTags could be increased for example by scaling all values between 0 and 1 instead of 0.4 and 1 and 0.125 and one, respectively. The value of a specific term could be emphasized for instance by multiplying it by a suitable scalar. Another factor that might improve the results would be to change the document similarity and the number of QTags into global values. At the moment, only the values provided for the the same question are comparable with each other. On the other hand, also the effect of changing the values of the context sizes into question-specific instead of global could have an effect on the results. However, all this experimentation with different coefficients and question-specific and global values remains an interesting theme for future research.

8.2 Analysis of the answers

Each answer has to be categorized as either right, wrong or inexact in the evaluation. However, the division of answers into the categories right and inexact is not easy for all questions, and this division does affect the results as inexact answers are not regarded as right in any case. In the following, we present three example questions taken from the datasets listed in appendices 1 and 2, the answers of which the reader may find quite inexact.

Example 8.1 *What is Eurostat? the EU's statistical office*

In Example 8.1, the word *Luxembourg-based* seems redundant. The following three text snippets are extracted from the document collection.

1. Eurostat, **the European Commission's statistics office**
2. **the statistical office**, known as Eurostat
3. Eurostat, **the European Union statistical office**

Text snippet number one appears twice in it. The answer provided by text snippet number two might be incomplete and thus inexact. An undoubtedly right answer could have been picked either from the text snippet number one or three. The answer candidates are marked in boldface in the text snippets.

Example 8.2 *What task does the French Academy have? fix rules of usage*

The answer to the question in Example 8.2 seems quite incomplete. The reader is tempted to ask: *Fix rules of usage for what?*. In the document collection, there is only one text snippet that is about the tasks of the French Academy. It is the following:

> All my law says is that, in certain cases, **French** must be used. The law doesn't **fix rules of usage**. That is the responsibility of the French Academy, the intellectuals and the popular users.

The possible answer candidate is marked in boldface. However, it could not be used as such because it is not a continuous text snippet and it should be reformulated into something such as *fix rules of usage for the French language*. Given the document collection, the answer *fix rules of usage* is quite right and not as inexact as it would seem at first sight.

Example 8.3 *Who chairs the European Parliament Committee on the Environment, Public Health, and Consumer Protection? Collins*

The answer to the question in Example 8.3 seems incomplete as well. The reader would expect at least the first name of the person in addition to his last name. Additional information such as the nationality of the person would not seem redundant either. In the following is the only text snippet in the collection that contains the answer to the question:

> SCOTTISH businesses will have nothing to fear from a European licensing system to ensure compliance with environmental regulation, if they are genuinely as compliant as they claim to be, the **Strathclyde East Euro-MP Ken Collins** told members of CBI Scotland yesterday. Mr Collins, who is chairman of the European Parliament Committee on the Environment, Public Health, and Consumer Protection,

The answer candidate that would be more complete and thus right is marked in boldface in the above text snippet. However, extracting it is not straightforward because then the method would need to make the anaphoric inference that *Collins* in the following sentence refers to the same person as the *Strathclyde East Euro-MP Ken Collins*.

As can be observed in the above examples, answer generation is challenging because the answers have to consist of text snippets directly taken from the document collection. No words may be skipped and left out and no new answer strings may be generated based on inferences made from the text. One might thus ask if the evaluation should take into account the difficulty of the question given the document collection used. In Example 8.1, the answer could be judged redundant and thus inexact because the collection contains many non-redundant or right answers. Along the same lines, the answer to the question in Example 8.2, which seems a bit incomplete, should not be judged as such because the document collection does not contain any more complete answers. The case of the answer for the question in Example 8.3 is a bit more debatable. However, as there exist good methods for the resolution of anaphoric references in English text (see e.g. [Mit03a]), one would expect that the system makes use of such a method and thus returning only *Collins* as the answer should be judged as inexact.

While evaluating the methods, it turned out that the methods found many answers that do not appear in the training or test data files listed in Appendices 1 and 2. The new answers found are shown in Figures 7.3, 7.4, 7.5 7.6 and 7.7. Each answer is marked with an R or an X showing whether it was judged right or inexact. As may be seen in these figures, the decisions are not always straightforward.

After discussing the difficulty of categorizing answers into right and inexact ones, it must be admitted that for some questions this categorization is quite straightforward. This type of questions specify very clearly the type of answer that is expected. Examples of such questions and of their expected answer types are:

Name a German car producer. A German car producer.
In which year did the Islamic revolution take place in Iran? A year.
What does "UAE" stand for? The non-abbreviated form of the acronym UAE.

The correctness of the answers is judged based on the document collections used and not based on any other knowledge. Thus, the answers returned by the system are regarded as being correct even if they actually are wrong because the document collection contains mistakes. In addition, all answers in the document collection are regarded as equal. For example, for the question *Who is Paul Simon*, the system answers *politician* and not *a singer and song maker*, and the answer provided by the system is regarded as equally good as if it had returned the latter one. The document collection used consists of newspaper text from the years 1994 and 1995, which is of course reflected in the answers, for example *When did the bomb attack at the World Trade Center occur? two years ago.*

8.3 Comparison with other methods

The main differences between the novel answer extraction method and the already existing ones are that most existing methods use a very fine-grained question classification and they only extract answers to a limited type of questions, such as questions of type *Who is . . . ?*. In addition, many existing methods require hand-crafted patterns. The use of a fine-grained question classification permits the introduction of very specified AEPs, which are potentially more precise. This is also often shown to be the case in practice as the results given in Section 3.1.3 show. However, this approach demands a very sophisticated question classifier module, and much of the work performed by the answer extraction module in our system would be moved into the question classification module. Systems that report experiments concerning only one or a very limited number of question types circumvent the need of a more complex question classification.

Another type of difference between the existing methods and the new ones is in the form of the patterns themselves. The new method uses very simple patterns: just part-of-speech tags, plain words, punctuation and capitalization patterns. In addition to these, most other methods use syntactic parsers, complex named entity taggers, recognizers for temporal and numeric expressions and thesauri. Due to the simplicity of the patterns, the new methods are more readily extensible to other languages where the existing tools might be different from the ones developed for English or where they may be missing altogether. However, when applying the new

method for a new language, the patterns might need slight modifications if the language differs significantly from the English language. For example, one could imagine that when using the method for Finnish, which is an agglutinative and synthetic language, one would benefit from using lemmas instead of plain words and from using information about morphological cases in addition to part-of-speech information.

The AEPs of the novel methods may contain arbitrarily many words from the question. Most existing methods may take only one question word into the AEP, if any. Thus, the new method makes more efficient use of the information present in the question than most other methods.

8.4 Limitations and future work

This section presents limitations of the present work and gives ideas for future work. First a simple method for reducing the processing time of the answer extraction methods introduced in the thesis is given. Subsequently, the impact of the given question classification to answer extraction is discussed. Methods for investigating different question classifications are given.

A simple method for reducing the processing time of the answer extraction method is presented in the following. The reduction of processing time is achieved using both the document similarity values and the answer scores. One limitation of the current method is that it is slow if both the number of patterns and the number of text snippets against which they are matched is high. This is the case especially in the alignment based method when the class is *Definition Person*, the number of QTags is 2 and the query matches many documents with a document similarity value greater than 0.4. This limitation could be quite easily resolved by only gradually decreasing the document similarity threshold values and by using the scores of the extracted answers to determine whether further processing is needed. In such a system, the document similarity threshold value would be quite high in the beginning. The *Answer Extractor* would take as input each preprocessed paragraph and its associated document similarity value, match the relevant patterns to the paragraphs and score the possible matches, or answer candidates. If an answer candidate with a sufficiently high score is found, the QA system would resume execution and return the candidate as the answer. If no such answer candidate is found after all relevant paragraphs have been matched, another document retrieval phase would be executed with a lower document similarity threshold value. If the final document similarity threshold value is reached and no answer is found, the

system would return the string *NIL*, which would indicate that it believes that the document collection contains no answer to the question. Thus, the search for answers would proceed in a greedy manner as not all candidates would be examined if a sufficiently good candidate is found. This would make the overall performance of the system more efficient. Naturally, the processing time of the system would not improve in all cases. It would stay the same in the case of NIL questions and in the cases where the system is not able to find good answer candidates from documents whose document similarity is higher than the document similarity threshold value. Determining the answer candidate score threshold value for stopping the processing could be performed quite straightforwardly in the same manner as the document similarity value has been determined i.e. by examining two answer candidate values in the training data: those representing right answers and those representing wrong answer candidates to NIL questions.

The effect of the question typology on the results of the answer extraction method is discussed in the following. All answer extraction is performed based on class-specific patterns. This presumes that the question classification reflects differences in the contexts of the answers. For example, it is assumed that the answer contexts for questions of class *Definition Person* and *Definition Organization* are different. The AEPs reflect directly the contexts of the answers. Tables 6.2 and 6.3 on pages 65 and 67, respectively, show that a typical AEP appears in only one class. In Section 6.2.1 on page 62 some very preliminary arguments suggesting that the classification adopted suits some classes better than others are put forward. The arguments are based on the fact that some classes contain more AEPs that belong solely to that class than other classes. However, the differences between classes are too small for making any definitive conclusions. At first sight, this would suggest that the answer contexts reflect the question classes well and that it is sensible to perform answer extraction using these class-specific pattern classes. However, no study has been made what kind of results a random partition of the questions into classes would have produced. There might also very well exist a human-made classification of the questions that would better reflect the similarity of the answer contexts inside one class than the classification presently used. One way to investigate this would be to perform a clustering of the answer patterns and to form question classes from the clusters formed.

If the AEPs belonging to a question class have no more in common with each other than any random AEPs, it makes no sense to perform the answer extraction based on these classes. On the other hand, if there exists a better way to classify the AEPs than the one currently used, it would

probably provide better answer extraction results.

In addition to the above-discussed two major limitations and directions for future work, several other interesting and potentially useful directions of work were also encountered. The most important ones consist of extending the method and experiments to other languages and other applications. Examples of interesting applications for which the method is suited are the semantic annotation of text and information extraction from text. However, these are out of the scope of this thesis.

Chapter 9

Conclusion

A textual question answering system is a system that seeks answers to natural language questions from unstructured text. Textual question answering systems are an important research problem because as the amount of natural language text in digital format grows all the time, the need for novel methods for pinpointing important knowledge from the vast text databases becomes more and more urgent. In addition to this, textual question answering systems form a well-defined framework with lots of existing evaluation data in which new methods can be developed and evaluated in a principled way. The separate subproblem of developing answer extraction methods for extracting answers from unstructured text is an interesting problem not only by itself but also because it is quite similar to many other problems, such as the problems of information extraction from text and of semantic annotation of text. Thus, the novel answer extraction methods may be generalized for these problems. As a matter of fact, the methods developed in this thesis may be expanded to any problem where the training data can be preprocessed into the required format. This format is quite general as it only consists of an input question and of its answer. Examples of problems that can be processed into this format are information extraction from text and semantic annotation of text. In the case of information extraction from text, the template to be filled can be expressed as a set of natural language questions. The same applies to semantic annotation. For example, in order to annotate semantically all names of films appearing in a text, the system is given as input both the question *Name a film.* and a list of films appearing in the text document collection that is to be used as the training data.

The main contributions of this thesis are the development and evaluation of both a new type of answer extraction pattern and of two different methods for their automatic generation. The pattern matching based ap-

proach is chosen because of its language and application independence and because pattern matching based techniques have become one of the most successful methods in textual question answering over the last years.

The answer extraction methods are developed in the framework of our own question answering system. Publicly available datasets in the English language are used as training and evaluation data for the methods. The methods developed are based on the well-known methods of sequence alignment and hierarchical clustering. The similarity metric used is based on edit distance.

The new answer extraction patterns developed consist of the most important words in the question, part-of-speech tags, plain words, punctuation marks and capitalization patterns. The two new methods for creating answer extraction patterns are named the concatenation based method and the alignment based method. The performance of the answer extraction patterns and of the methods for generating them is measured indirectly through the performance of the patterns in the answer extraction task of a question answering system. The difference in performance between the concatenation based and the alignment based answer extraction pattern generation methods is not important when evaluated using the evaluation data. However, when evaluated using the training data and when taking into account only the first answer candidate, the alignment based method performs better than the concatenation based one. The average accuracy of the question answering system when evaluated with the test data is approximately 0.17.

The main conclusions of the research are that answer extraction patterns consisting of the most important words of the question, plain words, part-of-speech tags, punctuation marks and capitalization patterns can be used in the answer extraction module of a question answering system. This type of patterns and the two new methods for generating them provide average results when compared to those produced by other systems using the same dataset. However, most answer extraction methods in the question answering systems using the same dataset are hand-crafted. The significance of the results obtained reside in the fact that the new methods require no manual creation of answer extraction patterns. As a source of knowledge, they only require a dataset of sample questions and answers, as well as a set of text documents that contain answers to most of the questions. Despite these results there is still plenty of future work to be done. The current methods could be made more efficient, the question classification could be analyzed and a classification more suitable for the novel method might be created, and the novel answer extraction patterns and methods

for their generation could be extended to other languages and applications – just to name a few very concrete steps that could be taken in the future.

The field of textual question answering is a field of active research, and its popularity has been growing all the time if one judges by the number of participating institutions in the NTCIR, CLEF and TREC question answering system evaluation campaigns. Future trends in textual question answering include shifting the focus to user interaction, efficiency issues and to more heterogeneous and voluminous text collections. User interaction in question answering has been introduced into TREC in the form of the complex interactive question answering track, which has been available since 2006. CLEF 2006 question answering evaluation track also included a pilot task that evaluated cross-language question answering systems in a real, user-oriented scenario [VMG+06]. Efficiency issues have been evaluated at CLEF since 2006 in the Real Time question answering Exercises. The growing interest in using more heterogeneous and voluminous text collections in textual question answering can be observed in the increasing tendency to exploit the World Wide Web in addition to the newspaper text collections provided by the organizers of the evaluation campaigns. In addition to this, the text database from which answers may be searched at CLEF 2007 consists of both the newspaper collections and of the Wikipedia. The future trends in answer extraction methods seem to be in line with the future trends of question answering. Data-driven pattern based methods for the generation of answer extraction patterns are likely to become more and more popular as the text databases from which the answers are sought become more massive and more heterogeneous. In addition to this, the efficiency of the methods is likely to become an important issue. Especially those parts of the methods that are used in an on-line manner by end users are required to perform promptly. Altogether, the future of textual question answering and answer extraction is likely to be very interesting.

References

[AHK+03] Lili Aunimo, Oskari Heinonen, Reeta Kuuskoski, Juha Makkonen, Renaud Petit, and Otso Virtanen. Question answering system for incomplete and noisy data: Methods and measures for its evaluation. In *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 193 – 206, Pisa, Italy, 2003.

[AK05] Lili Aunimo and Reeta Kuuskoski. Reformulations of Finnish questions for question answering. In *Proceedings of the 15th Nordic Conference of Computational Linguistics*, Joensuu, Finland, May 2005.

[ALM+04] Carlos Amaral, Dominique Laurent, André Martins, Afonso Mendes, and Cláudia Pinto. Design and implementation of a semantic search engine for Portuguese. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 247–250, Lisbon, Portugal, May 2004.

[AS05] Andrea Andrenucci and Eriks Sneiders. Automated question answering: Review of the main approaches. In *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2*, pages 514–519, Washington, DC, USA, 2005. IEEE Computer Society.

[Aya05] Christelle Ayache. Campagne d'evaluation EqueR-EVALDA. Evaluation en Question-Réponse. Technical report, Evaluations and Language Resources Distribution Agency (ELDA), 2005.

[BCW88] R. A. Becker, J.M. Chambers, and A. R. Wilks. *The New S Language: a programming environment for data analysis and graphics*. Wadsworth & Crooks/Cole, 1988.

[BHK+97] Robin Burke, Kristian Hammond, Vladimir Kulyukin, Steven Lytinen, Noriko Tomuro, and Scott Schoenberg. Question answering from frequently asked question files. *AI Magazine*, 18(2):57–66, 1997.

[BL02] Regina Barzilay and Lillian Lee. Bootstrapping lexical choice via multiple-sequence alignment. In *Proceedings of the 2002 Conference on Empirical Methods in Naural Language Processing*, pages 164–171, 2002.

[BL03] Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 16–23, 2003.

[BSA00] Stephan Busemann, Sven Schmeier, and Roman G. Arens. Message classification in the call center. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, Washington, USA, 2000.

[CCK+02] C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P.Tilker. Statistical selection of exact answers (multitext experiments for TREC 2002). In Voorhees and Buckland [VB02].

[Chi98] Nancy A. Chinchor. Overview of MUC-7/MET-2. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Virginia, USA, April 1998.

[CLE06] Question answering at CLEF 2006. Guidelines for participants. http://clef-qa.itc.it/DOWNLOADS/QA@CLEF Guidelines for Participants.pdf, 2006. Final version - 060327.

[CR97] Nancy Chinchor and D. Robinson. MUC-7 named entity task definition (version 3.5). In *Proceedings of the 7th Message Understanding Conference*, September 1997.

[Dam64] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Communications of the Association for Computing Machinery*, 7(3):171–176, 1964.

[DBB+02] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, 2002.

[DEKM98] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[EM03] A. Echihabi and D. Marcu. A Noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003.

[Fel98] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[FHE03] Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[FKM03] Junichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question answering challenge (QAC-1): An evaluation of question answering task at NTCIR Workshop 3. In *Proceedings of the NTCIR Workshop 3*, 2003.

[FKM04a] Junichi Fukumoto, Tsuneaki Kato, and Fumito Masui. An evaluation of question answering challenge (QAC-1) at the NTCIR Workshop 3. *SIGIR Forum*, 38(1):25–28, 2004.

[FKM04b] Junichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question answering challenge for five ranked answers and list answers – overview of NTCIR4 QAC-2 subtask 1 and 2. In *Proceedings of the NTCIR Workshop 4*, 2004.

[Gad88] T. N. Gadd. 'Fisching fore Werds': Phonetic retrieval of written text in information systems. *Program: Automated Library and Information Systems*, 22(3):222–237, 1988.

[GCL61] W. Green, C. Chomsky, and K. Laugherty. Baseball: An automatic question answerer. In *Proceedings of the Western Joint Computer Conference*, pages 219–224, 1961.

[Gus97] Dan Gusfield. *Algorithms on Strings, Trees and Sequences. Computer Science and Computational Biology.* Press Syndicate of the University of Cambridge, 1997.

[GW98] R. Gaizauskas and Y. Wilks. Information extraction: beyond document retrieval. *Journal of Documentation*, 54(1):70–105, 1998.

[GW06] Shlomo Geva and Alan Woodley. The NLP task at INEX 2005. *SIGIR Forum*, 40(1):60–63, 2006.

[HD80] Patrick A. V. Hall and Geoff R. Dowling. Approximate string matching. *Computing Surveys*, 12(4):381–402, 1980.

[HEM02] Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. Natural language based reformulations resource and web exploitation for question answering. In Voorhees and Buckland [VB02].

[HG01] Hirschman and Gaizauskas. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300, 2001.

[HG04] Erik Hatcher and Otis Gospodnetić. *Lucene in Action.* Manning Publications Co., 2004.

[HHR02] Eduard Hovy, Ulf Hemjacob, and Deepak Ravichandran. Question/answer typology with surface text patterns. In *Proceedings of the Human Language Technology Conference*, San Diego, USA, 2002.

[HM03] Sanda Harabagiu and Dan Moldovan. Question Answering. In Mitkov [Mit03b].

[HMC+05] Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, Andrew Hickl, and Patrick Wng. Employing two question answering systems in TREC-2005. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, Maryland, United States, November 2005.

[HMP+01] Sanda M. Harabagiu, Dan I. Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *The 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 274–281, 2001.

[HPV05] Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. Question answering pilot task at CLEF 2004. In Peters et al. [PCJ+05].

[JdRM04] Valentin Jijkoun, Maarten de Rijke, and Jori Mur. Information extraction for question answering: Improving recall through syntactic patterns. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, 2004.

[JGTVDC+06] Antonio Juárez-Gonzalez, Alberto Téllez-Valero, Claudia Denicia-Carral, Manuel Montes y Gómez, and Luis Villaseñor-Pineda. INAOE at CLEF 2006: Experiments in spanish question answering. In *Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, September 2006.

[JMdR03] Valentin Jijkoun, Gilad Mishne, and Maarten de Rijke. Preprocessing documents to answer Dutch questions. In *Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'03)*, 2003.

[Jon72] Karen Sparck Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[JT97] Timo Järvinen and Pasi Tapanainen. A dependency parser for English. Technical Report TR–1, Department of General Linguistics, University of Helsinki, 1997.

[KL03] Boris Katz and Jimmy Lin. Selectively using relations to improve precision in question answering. In *Proceedings of the EACL-2003 Workshop on Natural Language Processing for Question Answering*, April 2003.

[Kru83] Joseph B. Kruskal. An overview of sequence analysis. In David Sankoff and Joseph B. Kruskal, editors, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1966.

[Lin05] Linguistic Data Consortium. *ACE (Automatic Content Extraction) English Annotation Guidelines for Entities*, version 5.6.1 2005.05.23 edition, May 2005.

[LK03] Jimmy Lin and Boris Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the twelfth international conference on Information and knowledge management (CIKM'03)*, pages 116–123, New York, NY, USA, 2003. ACM Press.

[LK05] Jimmy Lin and Boris Katz. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 2005.

[LSN06] Dominique Laurent, Patrick Seguela, and Sophie Negre. Cross lingual question answering using QRISTAL at CLEF 2006. In *Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, September 2006.

[Mit03a] Ruslan Mitkov. Anaphora resolution. In *The Oxford Handbook of Computational Linguistics* [Mit03b].

[Mit03b] Ruslan Mitkov, editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.

[MMM+06] Andre Martins, Afonso Mendez, Pedro Mendes, Claudia Pinto, and Daniel Vidal. Priberam's question answering system in a cross-language environment. In *Working Notes for the CLEF 2006 Workshop*, Alicante, Spain, September 2006.

[Mon03] Christof Monz. *From Document Retrieval to Question Answering*. PhD thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, December 2003.

[MRV+04] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Victor Peinado, Felisa Verdejo, and Maarten de Rijke. The Multiple Language Question Answering Track at CLEF 2003. In Carol Peters, Julio Gonzalo, Martin Braschler, and Michael Kluck, editors, *Comparative Evaluation of Multilingual Information Access Systems. Fourth Workshop of the Cross-Language Evaluation Forum, CLEF 2003, Trondheim, Norway, August 2003. Revised papers*, volume 3237 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.

[MSM93] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[MV+05] B. Magnini, A. Vallin, , L. Aunimo, C.Ayache, G. Erbach, A. Penas, M. de Rijke, P. Rocha, K. Simov, and R. Sutcliffe. Overview of the CLEF 2005 Multilingual Question Answering Track. In Carol Peters and Francesca Borri, editors, *Proceedings of the CLEF 2005 Workshop*, Vienna, Austria, September 2005.

[MVA+05] Bernardo Magnini, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas, Maarten de Rijke, Paulo Rocha, Kiril Simov, and Richard Sutcliffe. Overview of the CLEF 2004 Multilingual Question Answering Track. In Peters et al. [PCJ+05].

[PCJ+05] C. Peters, P. D. Clough, G. J. F. Jones, J. Gonzalo, M. Kluck, and B. Magnini, editors. *Multilingual Information Access for Text, Speech and Images: Results of*

the Fifth CLEF Evaluation Campaign, CLEF 2004, Bath, United Kingdom, Revised Selected Papers, volume 3491 of *Lecture Notes in Computer Science*. Springer Verlag, November 2005.

[Pet06] Carol Peters. What happened in CLEF 2005? In Peters et al. [PGG+06].

[PGG+06] Carol Peters, Fredric C. Gey, Julio Gonzalo, Gareth J.F.Jones, Michael Kluck, Bernardo Magnini, Henning Mueller, and Maarten de Rijke, editors. *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005, Vienna, Austria, Revised Selected Papers*, volume 4022 of *Lecture Notes in Computer Science*. Springer Verlag, 2006.

[Por80] M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

[Por01] M.F. Porter. Snowball: A Language for Stemming Algorithms, 2001. Available at http://snowball.tartarus.org/texts/introduction.html [7.2.2007].

[RH02] Deepak Ravichandran and Edouard Hovy. Learning surface text patterns for a question answering system. In *the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 41–47, Philadelphia, PA., 2002.

[Sal71] Gerard Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[SChCL05] Yutaka Sasaki, Hsin-Hsi Chen, Kuang hua Chen, and Chuan-Jie Lin. Overview of the NTCIR-5 cross-lingual question answering task (CLQA-1). In *Proceedings of the NTCIR Workshop 5*, 2005.

[SI99] Satoshi Sekine and Hitoshi Isahara. IREX Information Retrieval and Extraction Exercise project overview. In *Proceedings of the Information Retrieval and Extraction Exercise (IREX) Workshop*, 1999.

[Sim65] R. Simmons. Answering English questions by computer: A survey. *Communications of the ACM*, 8(1):53–70, 1965.

[SS01] Martin M. Soubbotin and Sergei M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In Voorhees and Harman [VH01].

[SS02] Martin M. Soubbotin and Sergei M. Soubbotin. Use of patterns for detection of answer strings: A systematic approach. In Voorhees and Buckland [VB02].

[SS05] Peter Siniakov and Christian Siefkes. An overview and classification of adaptive approaches to information extraction. *Journal on Data Semantics*, IV:172–212, 2005.

[TL04] Noriko Tomuro and Steven Lytinen. Retrieval models and Q & A learning with FAQ files. In Mark T. Maybury, editor, *New Directions in Question Answering*, pages 183–194. AAAI Press, 2004.

[TPK+05] Jarmo Toivonen, Ari Pirkola, Heikki Keskustalo, Kari Visala, and Kalervo Järvelin. Translating cross-lingual spelling variants using transformation rules. *Information Processing and Management*, 41(4):859–872, 2005.

[VB02] E. M. Voorhees and Lori P. Buckland, editors. *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology, 2002.

[VD05] Ellen M. Voorhees and Hoa Trang Dang. Overview of the TREC 2005 question answering track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, Maryland, United States, November 2005.

[VH01] E. M. Voorhees and D. K. Harman, editors. *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2002)*. Department of Commerce, National Institute of Standards and Technology, 2001.

[VMG+06] Alessandro Vallin, Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Peya Osenova, Anselmo Peñas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, and Richard Sutcliffe. Overview of the CLEF 2005

Multilingual Question Answering Track. In Peters et al. [PGG+06].

[Voo99] Ellen M. Voorhees. The TREC-8 Question Answering Track Report. In Ellen M. Voorhees and D. K. Harman, editors, *Proceedings of TREC-8*, Gaithersburg, Maryland, November 1999. Department of Commerce, National Institute of Standards and Technology.

[Voo02] Ellen M. Voorhees. Overwiew of the TREC-2002 Question Answering Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of TREC-2002*, Gaithersburg, Maryland, November 2002. Department of Commerce, National Institute of Standards and Technology.

[Voo04] Ellen M. Voorhees. Overview of the trec 2004 question answering track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, Maryland, United States, November 2004.

[Vos98] Piek Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers, Dordrecht, 1998.

[vR75] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 1975.

[vZ01] Menno van Zaanen. *Bootstrapping Structure into Language: Alignment-Based Learning*. PhD thesis, The University of Leeds, School of Computing, September 2001.

[WF74] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, 1974.

[WG04] Alan Woodley and Shlomo Geva. NLPX at INEX 2004. In *Proceedings of the Initiative for the Evaluation of XML Retrieval (INEX) 2004 Workshop*, Dagstuhl Castle, Germany, 2004.

[Woo73] William A. Woods. Progress in natural language understanding: an application to lunar geology. In *AFIPS Conference Proceedings, 42*, pages 441–50, New York, June 1973. American Federation of Information Processing Societies.

[XWL04] Jinxi Xu, Ralph Weischedel, and Ana Licuanan. Evaluation of an extraction-based approach to answering definitional questions. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 418–424, New York, NY, USA, 2004. ACM Press.

[YK06] Jamileh Yousefi and Leila Kosseim. Automatic acquisition of semantic-based question reformulations for question answering. In *7th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2006*, pages 441–452, Mexico City, Mexico, February 2006.

[ZD96] Justin Zobel and Philip Dart. Phonetic string matching: Lessons from information retrieval. In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pages 166–172, Zurich, Switzerland, 1996.

Appendix 1. Questions and answers of the training data set. Answers are annotated with the MUC-7 named entity task categories. The annotations are only used in the analysis of the answer data.

ORGANIZATION D What is UNITA? the <ENAMEX TYPE="ORGANIZATION"> National Union for the Total Independence of Angola </ENAMEX>
ORGANIZATION D What is the FARC? <ENAMEX TYPE="ORGANIZATION"> Revolutionary Armed Forces of Colombia</ENAMEX>
PERSON D Who is Javier Solana? the Spanish Foreign Minister
PERSON D Who was Yasser Arafat? <ENAMEX TYPE="ORGANIZATION"> Palestine Liberation Organization</ENAMEX> Chairman
PERSON D Who is Giulio Andreotti? former Italian Prime Minister
LOCATION F Where is the Brandenburg Gate? <ENAMEX TYPE="LOCATION"> Berlin</ENAMEX>
LOCATION F Name a building wrapped by Christo. <ENAMEX TYPE="LO CATION"> Reichstag</ENAMEX>
LOCATION F Where is the ozone hole? <ENAMEX TYPE="LOCATION"> Antarctic</ENAMEX>
LOCATION F Name a city in which Giulio Andreotti was put on trial. <ENAMEX TYPE="LOCATION"> Palermo</ENAMEX>
MEASURE F How old is Jacques Chirac? 62
ORGANIZATION F Name a German car producer. <ENAMEX TYPE="ORGANIZATION"> Volkswagen</ENAMEX>
LOCATION F Name a country which borders on the Kaliningrad enclave. <ENAMEX TYPE="LOCATION"> Poland</ENAMEX>
LOCATION F Name one of the seven wonders of the world. <ENAMEX TYPE="LOCATION"> Pharos Lighthouse</ENAMEX>
OTHER F Name a film in which computer animation was used. The Pagemaster
PERSON F Who played the role of Superman before being paralyzed? <ENAMEX TYPE="PERSON"> Christopher Reeve</ENAMEX>
PERSON F Who is the founder of Greenpeace? <ENAMEX TYPE="PERSON"> David McTaggart</ENAMEX>
PERSON F Who is the head of the FSB? <ENAMEX TYPE="PERSON"> Sergei Stepashin</ENAMEX>
TIME F In which year did the Islamic Revolution take place in Iran? <TIMEX TYPE="DATE"> 1979</TIMEX>

TIME F When was Cyprus divided into two parts? <TIMEX TYPE= "DATE"> 1974</TIMEX>
TIME F In which year did Prince Charles marry Diana? <TIMEX TYPE= "DATE"> 1981</TIMEX>
TIME F Since when has the Dalai Lama been living in exile? <TIMEX TYPE="DATE"> 1959</TIMEX>
PERSON F Which professor from Bonn received the Nobel Prize for Economics? <ENAMEX TYPE="PERSON"> Reinhard Selten</ENAMEX>
TIME F When was the beginning of the European Monetary Union? <TI MEX TYPE="DATE"> January 1, 1999</TIMEX>
TIME F When did the Olympic Games take place in Atlanta? <TIMEX TYPE="DATE"> 1996</TIMEX>
MEASURE F How many victims of the massacres in Rwanda were there? between a half-million and 1 million
ORGANIZATION D What is the IFP? <ENAMEX TYPE="ORGANIZA TION"> Inkatha Freedom Party</ENAMEX>
PERSON D Who is Goodwill Zwelithini? Zulu King
ORGANIZATION D What is the Camorra? <ENAMEX TYPE="LOCA TION"> Naples</ENAMEX> Mafia
PERSON D Who is Antonio Di Pietro? <ENAMEX TYPE="LOCA TION"> Milan</ENAMEX> magistrate
MEASURE F How much money did Selten, Nash and Harsanyi receive for the Nobel Prize for Economics?<NUMEX TYPE="MONEY"> $930,000 </NUMEX>
ORGANIZATION D What is Eurostat? the <ENAMEX TYPE="ORGA NIZATION"> EU</ENAMEX> 's <ENAMEX TYPE="LOCATION"> Luxem bourg</ENAMEX> -based statistical office
OTHER F What task does the French Academy have? fix rules of usage
PERSON F Who chairs the European Parliament Committee on the Environment, Public Health, and Consumer Protection? <ENAMEX TYPE= "PERSON"> Collins</ENAMEX>
PERSON F Who takes the final decision on whether to authorize altar girls in dioceses? local bishops
PERSON F Which two scientists discovered "G proteins"? <ENAMEX TYPE="PERSON"> Alfred G. Gilman</ENAMEX> ,53 , of the <ENA MEX TYPE="ORGANIZATION"> University of Texas Southwestern Medical Center</ENAMEX> in <ENAMEX TYPE="LOCATION"> Dallas </ENAMEX> and <ENAMEX TYPE="PERSON"> Martin Rodbell </ENAMEX>
PERSON F Who planted the first known vineyard in the world? <ENA

MEX TYPE="PERSON"> Noah</ENAMEX>
PERSON F Who wrote: "of all the animals, man is the only one that is cruel"? <ENAMEX TYPE="PERSON"> Mark Twain</ENAMEX>
PERSON F Who was Haiti's former military commander in chief? Lt. Gen. <ENAMEX TYPE="PERSON"> Raoul Cedras</ENAMEX>
PERSON F Who is El Nino named after? the <ENAMEX TYPE="PER SON"> Christ</ENAMEX> child
PERSON F Who were the two signatories to the peace treaty between Jordan and Israel? <ENAMEX TYPE="PERSON"> Hussein</ENA MEX> and <ENAMEX TYPE="PERSON"> Rabin</ENAMEX>
PERSON F Whose nickname was "the Balkan Kissinger"? <ENA MEX TYPE="PERSON"> Holbrooke</ENAMEX>
PERSON F Who was the last king of Iraq? <ENAMEX TYPE="PER SON"> Faisal</ENAMEX>
OTHER F What do supporters of the Zapatistas in Mexico wear? masks
PERSON F Who ended a government ban on "100 percent" foreign ownership of companies in Japan in 1973? Prime Minister <ENAMEX TYPE="PERSON"> Kakuei Tanaka</ENAMEX>
PERSON F Who was the President of the United States of America between 1976 and 1980? <ENAMEX TYPE="PERSON"> Carter </ENA MEX>
PERSON F Who praised Indurain for his "capacity for sacrifice, class and healthy spirit of competition", when he won the Tour de France for the fourth time? Spanish Prime Minister <ENAMEX TYPE="PERSON"> Felipe Gonzalez</ENAMEX>
PERSON F Who took the decision to send tanks to Chechnya on 11 December? President <ENAMEX TYPE="PERSON"> Boris Yeltsin </ENA MEX>
PERSON F Who continued to be supplied with contaminated blood products six months after an American blood-screening test and heat-treatment process were available in other countries? French hemophiliacs
PERSON D Who was Charles Bukowski? writer and poet laureate
PERSON D Who is Radwa Ashour? a <ENAMEX TYPE="LOCA TION"> Cairo</ENAMEX> novelist and professor
PERSON D Who is Rolf Ekeus? the <ENAMEX TYPE="ORGANIZA TION"> UN</ENAMEX> 's chief arms inspector
PERSON D Who is Dzhokhar Dudayev? Chechen leader
PERSON D Who is Willy Claes? <ENAMEX TYPE="ORGANIZA TION"> Nato</ENAMEX> secretary-general
PERSON D Who was Emiliano Zapata? revolutionary hero

PERSON D Who is Antonio Matarrese? <ENAMEX TYPE="ORGANIZATION"> Italian Soccer Federation</ENAMEX> president
PERSON D Who is Vladimir P. Melnikov? scientist
PERSON D Who is Uffe Ellemann-Jensen? former Danish Foreign Minister
PERSON D Who was Pibul? <ENAMEX TYPE="LOCATION"> Thailand </ENAMEX> 's prime minister
PERSON D Who is Yasushi Akashi? <ENAMEX TYPE="ORGANIZATION"> UN</ENAMEX> special envoy
PERSON D Who was Andrei Kozyrev? Russian Foreign Minister
TIME F What time of year does El Nino usually begin? <TIMEX TYPE="DATE"> mid to late December</TIMEX>
TIME F Since when have Israel and Jordan been in a state of war? <TIMEX TYPE="DATE"> since 1948</TIMEX>
TIME F When did Gorbachev become the "laughing stock of the nation"? in <TIMEX TYPE="DATE"> 1991</TIMEX>
TIME F When was the treaty on Conventional Forces in Europe signed? <TIMEX TYPE="DATE"> 1990</TIMEX>
TIME F On which date is winter assumed to start in Bosnia? <TIMEX TYPE="DATE"> October 15</TIMEX>
TIME F In which decade did Japanese auto makers invest heavily? <TIMEX TYPE="DATE"> in the late 1980s</TIMEX>
TIME F In which year was Baggio the world soccer player of the year? <TIMEX TYPE="DATE"> 1993</TIMEX>
TIME F When will the Human Genome Project be completed? <TIMEX TYPE="DATE"> 2005</TIMEX>
TIME F When did Genghis Khan die? <TIMEX TYPE="DATE"> 1227 </TIMEX>
LOCATION F Which countries form the world's largest and richest consumer market? The union's member nations
LOCATION F Name a traffic free resort in Switzerland. <ENAMEX TYPE="LOCATION"> Saas Fee</ENAMEX>
LOCATION F From which port did the ferry Estonia begin its last voyage? <ENAMEX TYPE="LOCATION"> Tallinn</ENAMEX>
LOCATION F Which Russian city is twinned with Glasgow? <ENAMEX TYPE="LOCATION"> Rostov-on-Don</ENAMEX>
LOCATION F Which two German cities are connected to Hamburg by high-speed express trains? <ENAMEX TYPE="LOCATION"> Frankfurt </ENAMEX> and <ENAMEX TYPE="LOCATION"> Munich</ENAMEX>

LOCATION F What was Belarus previously called? <ENAMEX TYPE= "LOCATION"> White Russia</ENAMEX>

LOCATION F In which country is euthanasia permitted if requested by a patient suffering intolerable physical or mental pain? <ENAMEX TYPE= "LOCATION"> the Netherlands</ENAMEX>

LOCATION F Which countries can you travel to with the Scanrail 55+ pass? <ENAMEX TYPE="LOCATION"> Denmark</ENAMEX>, <EN AMEX TYPE="LOCATION"> Finland</ENAMEX>, <ENAMEX TY PE="LOCATION"> Norway</ENAMEX> and <ENAMEX TYPE="LO CATION"> Sweden </ENAMEX>

LOCATION F Which country returned thousands of sets of military remains in 1954? <ENAMEX TYPE="LOCATION"> North Korea </EN AMEX>

ORGANIZATION F Which corporation offers insurance and financing for American investments in Russia? <ENAMEX TYPE="ORGANI ZATION"> Overseas Private Investment Corp.</ENAMEX>

ORGANIZATION F Which company led the consortium that signed an agreement to invest in the basin on China's western border? <ENAMEX TYPE="ORGA NIZATION"> Agip</ENAMEX>

ORGANIZATION F Which US Army Division provided the paratroopers who took part in the invasion of Haiti? <ENAMEX TYPE="ORGA NIZATION"> 82nd Airborne</ENAMEX>

ORGANIZATION F Which Russian TV station is most sympathetic towards the Government? <ENAMEX TYPE="LOCATION"> Russia </ENAMEX> 's <ENAMEX TYPE="ORGANIZATION">Ostankino television </ENAMEX>

ORGANIZATION F For which Russian institution has Chechnya been a humiliation? <ENAMEX TYPE="ORGANIZATION"> Red Army </EN AMEX>

ORGANIZATION F Which institution demanded the immediate withdrawal of Bosnian Serb forces from Srebrenica? <ENAMEX TYPE="OR GANIZATION"> the United Nations Security Council </ENAMEX>

ORGANIZATION F Which oil company was accused by the Russian media of covering up a large oil spill in Siberia? <ENAMEX TYPE="OR GANIZATION"> Komineft</ENAMEX>

OTHER F Which computer virus was confirmed as a hoax by the US National Computer Security Association? Good Times

OTHER F Which symbol has been used to hallmark sterling silver in Scotland since 1473? the lion rampant

ORGANIZATION F Which Japanese car company lost $1 billion in

1993? <ENAMEX TYPE="ORGANIZATION"> Nissan</ENAMEX>
ORGANIZATION D What is Eurocare? an alliance of 23 alcohol abuse agencies within the <ENAMEX TYPE="ORGANIZATION"> European Union</ENAMEX>
ORGANIZATION D What is the Natural Resources Defense Council? an activist environmental group
ORGANIZATION D What is the OSCE? <ENAMEX TYPE="ORGA NIZATION"> Organisation of Security and Co-operation in Europe</EN AMEX>
ORGANIZATION D What is the WEU? the embryo defence arm of the <ENAMEX TYPE="ORGANIZATION"> EU</ENAMEX>
ORGANIZATION D What is the PRI? the <ENAMEX TYPE="ORGA NIZATION"> Institutional Revolutionary Party</ENAMEX>
ORGANIZATION D What is the Civic Alliance? An independent watchdog group
ORGANIZATION D What is the French SCPC? the official body responsible for sniffing out cases of administrative corruption
ORGANIZATION D What is Aum Shinrikyo? cult
ORGANIZATION D What was the GATT? General Agreement on Tariffs and Trade
ORGANIZATION D What is the WWF? <ENAMEX TYPE="OR GANIZATION"> World Wide Fund for Nature</ENAMEX>
ORGANIZATION D What is Shell? <ENAMEX TYPE="LOCA TION"> Europe</ENAMEX> 's biggest oil company
ORGANIZATION D What is the EZLN? <ENAMEX TYPE="OR GANIZATION"> Zapatista Army of National Liberation</ENAMEX>
MEASURE F What percentage of nuns in the Catholic Church are in favour of ordinating women?<NUMEX TYPE="PERCENT"> 57 percent </NUMEX>
MEASURE F How many people are diagnosed as suffering from colon cancer each year? 156,000
MEASURE F How far from Finland did the ferry Estonia sink? 23 miles southeast of the Finnish island of <ENAMEX TYPE="LO CATION"> Uto</ENAMEX>
MEASURE F How much will a European seed potato development project in Russia cost? <NUMEX TYPE="MONEY"> $60m</NUMEX>
MEASURE F How often does El Nino occur? at seven- to 12-year intervals
MEASURE F What percentage of children do not have enough food to eat in Iraq? <NUMEX TYPE="PERCENT"> between 22 and 30 percent

</NUMEX>
MEASURE F How many countries joined the international coalition to restore democratic government in Haiti? 25 nations
OTHER F What is responsible for most of the ecological disasters in Western Siberia? ruptured pipelines and leaking reservoirs
MEASURE F How big was the Siberian oil spill according to environmentalists? 200,000 tons
MEASURE F By how much have olive oil imports to the USA increased over the last ten years? <NUMEX TYPE="PERCENT"> 3,000 percent </NUMEX>
MEASURE F By how much did Japanese car exports fall between 1993 and 1994? <NUMEX TYPE="PERCENT"> 18.3 percent</NUMEX>
MEASURE F How big was the Japanese share of the American car market in 1992? <NUMEX TYPE="PERCENT"> 30.1 percent</NUMEX>
MEASURE F How many World Cup tournaments had Zagalo won as a player before Ronaldo was born in 1977? two
MEASURE F How many scandals was Tapie implicated in, while boss at Marseille? four
OTHER F Which was the first gene mutation responsible for breast cancer to be discovered? BRCA1
OTHER F Which enzyme acts on sugar to produce carbon dioxide and ethanol? zymase
OTHER F What was the nationality of most of the victims when the Estonia ferry sank? Swedish
OTHER F Which pesticide found in children's food has been condemned in a report? Alar
PERSON D Who is Simon Wiesenthal? <ENAMEX TYPE="ORGANI ZATION"> Nazi</ENAMEX> hunter
PERSON D Who is Jacques Blanc? the right-wing French <ENAMEX TYPE="ORGANIZATION"> CoR</ENAMEX> president
ORGANIZATION D What is Doctors Without Borders? French relief agency
ORGANIZATION D What is the UNAMIR? <ENAMEX TYPE="OR GANIZATION"> U.N. Assistance Mission in Rwanda</ENAMEX>
PERSON F Which Russian president attended the G7 meeting in Naples? <ENAMEX TYPE="PERSON"> Boris Yeltsin</ENAMEX>
PERSON F Who is the Norwegian king? <ENAMEX TYPE="PER SON"> Harald V</ENAMEX>
PERSON F Which French president inaugurated the Eurotunnel? <ENA MEX TYPE="PERSON"> Francois Mitterrand</ENAMEX>

PERSON F Who discovered Tutankhamun's tomb? <ENAMEX TYPE= "PERSON"> Howard Carter</ENAMEX>

LOCATION F Which city is the Al Aqsa Mosque in? <ENAMEX TYPE= "PERSON"> Jerusalem</ENAMEX>

LOCATION F What country does North Korea border on? <ENAMEX TYPE="PERSON"> China</ENAMEX>

MEASURE F How far is Jupiter from the Earth? 480 million miles

OTHER F What disease name does the acronym RSI stand for? Repetitive Strain Injury

OTHER F What vitamins help in the fight against cancer? C and E

PERSON F Who was the Norwegian Prime Minister when the referendum on Norway's possible accession to the EU was held? <ENAMEX TYPE="PERSON"> Gro Harlem Brundtland</ENAMEX>

PERSON F Who was Uganda's President during Rwanda's war? <ENA MEX TYPE="PERSON"> Yoweri Museveni</ENAMEX>

ORGANIZATION F Which terrorist group shot mortars during the attack on Heathrow Airport? <ENAMEX TYPE="ORGANIZATION"> IRA</ENAMEX>

PERSON F What minister was Silvio Berlusconi prior to his resignation? Premier

PERSON D Who is Umberto Bossi? leader of the <ENAMEX TYPE="OR GANIZATION"> Northern League</ENAMEX>

PERSON D Who is Joao Havelange? <ENAMEX TYPE="ORGANIZA TION"> FIFA</ENAMEX> 's Brazilian president

PERSON D Who is Richard Moller Nielsen? coach of <ENAMEX TYPE= "LOCATION"> Denmark</ENAMEX>

PERSON F What is Jari Litmanen's profession? striker

OTHER F What matters was Erkki Liikanen in charge of in the European Commission? budget

PERSON D Who is Flavio Briatore? <ENAMEX TYPE="PERSON"> Schumacher</ENAMEX> 's team chief

ORGANIZATION D What kind of an organization is Hamas? militant Muslim

ORGANIZATION D What is WAFA? <ENAMEX TYPE="ORGANIZA TION"> PLO</ENAMEX> news agency

LOCATION F What country is Fiat from? <ENAMEX TYPE="LOCA TION"> Italy</ENAMEX>

PERSON F What is Alexander Solzhenitsyn's wife's name? <ENAMEX TYPE="PERSON"> Natalia</ENAMEX>

PERSON F Who directed the film Condition Red? <ENAMEX TYPE= "PERSON"> Mika Kaurismaki</ENAMEX>
ORGANIZATION F What agency did Aldrich Ames work for? <ENA MEX TYPE="ORGANIZATION"> CIA</ENAMEX>
ORGANIZATION F What party does Roberto Maroni represent? <ENA MEX TYPE="ORGANIZATION"> Northern League</ENAMEX>
ORGANIZATION F What party does Edouard Balladur represent? conservative
ORGANIZATION F Which car company manufactures the Taurus? <ENAMEX TYPE="ORGANIZATION"> Ford</ENAMEX>
LOCATION F What is the capital of Cyprus? <ENAMEX TYPE="LO CATION"> Nicosia</ENAMEX>
LOCATION F Which F1 track was Ayrton Senna killed on? <ENAMEX TYPE="LOCATION"> Imola</ENAMEX>
OTHER F What kind of batteries does Volvo use? Lithium
OTHER F What award did Pulp Fiction win at the Cannes Film Festival? Palme d'Or
PERSON D Who was Kurt Cobain? <ENAMEX TYPE="ORGANIZA TION"> Nirvana</ENAMEX> lead singer
MEASURE F How many months did Luis Roldan's flight last? 10
TIME F When do we estimate that the Big Bang happened? 15 billion years ago
MEASURE F How many countries have ratified the United Nations convention adopted in 1989? 170
MEASURE F How many states are members of the Council of Europe? 35
LOCATION F Where will the Olympic Games take place in 2000? <ENA MEX TYPE="LOCATION"> Sydney</ENAMEX>
ORGANIZATION D What is Oxygen Software? NIL
MEASURE F What is the probability of children committing suicide before puberty? NIL
MEASURE F How much is the fine for speaking on a mobile phone when driving? NIL
MEASURE F How much did the USA pay for the Gulf War? NIL
MEASURE F How many people live in France? NIL
ORGANIZATION F Who is the main organizer of the international contest "Queen of the Future"? NIL
ORGANIZATION F What party did Andrei Brezhnev found? NIL
ORGANIZATION F According to which government did radioactivity from Chernobyl stop at the Franco-German border? NIL

OTHER F What newspaper was founded in Kiev in 1994? NIL
OTHER F Which space probe set off for the Moon on 25 January 1994? NIL
PERSON F Who became the Prime Minister of Slovenia in 1992? NIL
ORGANIZATION D What is the SLPN? NIL
MEASURE F How many communities did Di Mambro found? NIL
OTHER F Which EU conference adopted Agenda 2000 in Berlin? NIL
ORGANIZATION D What is the IFJ? NIL
OTHER F Which contract runs from 1995 to 2004? NIL
MEASURE F How many millions of people escaped from Eastern Europe into the FRG between 1950 and 1992? NIL
TIME F During which years was Samir Geagea a warlord in the Lebanese civil war? NIL
TIME F When was the safety zone in southern Lebanon created? NIL
TIME F What year did the president of Cyprus, Makarios III, die? NIL

Appendix 2. Questions and answers of the test data set. Answers are annotated with the MUC-7 named entity task categories. The annotations are used in the analysis of the answer data.

TIME F What year was Thomas Mann awarded the Nobel Prize? <TIMEX TYPE="DATE"> 1929</TIMEX>
PERSON F Who is the managing director of FIAT? <ENAMEX TYPE= "PERSON"> Cesare Romiti</ENAMEX>
ORGANIZATION F What party did Hitler belong to? < ENAMEX TYPE="ORGANIZATION"> Nazi</ENAMEX>
ORGANIZATION F What car company produces the "Beetle"? <ENA MEX TYPE="ORGANIZATION"> Volkswagen </ENAMEX>
OTHER F What is the Antarctic continent covered with? ice-cap
OTHER F What is the ozone hole caused by? man-made chlorine
TIME F When did the Ebola virus first appear? <TIMEX TYPE="DA TE"> 1976</TIMEX>
OTHER F Name a pesticide. DDT
TIME F On which day does Chinese New Year's Day fall? <TIMEX TYPE="DATE"> Feb. 10</TIMEX>
PERSON D Who is Jean-Bertrand Aristide? <ENAMEX TYPE="LOCA TION"> Haiti</ENAMEX> 's first democratically elected president
LOCATION F Where does El Nino occur? in the <ENAMEX TYPE="LO CATION"> Pacific</ENAMEX>
TIME F When did Nixon resign? on <TIMEX TYPE="DATE"> Aug. 9th, 1974</TIMEX>
ORGANIZATION F Who was the embargo against Iraq imposed by? The <ENAMEX TYPE="ORGANIZATION"> U.N. Security Council</ENA MEX>
OTHER F Name a cetacean. whale
PERSON F Who are the Zapatistas named after? <ENAMEX TYPE="PER SON"> Emiliano Zapata</ENAMEX>
MEASURE F How many human genes are there? 100,000
ORGANIZATION D Who are the Amish? descendants of the Swiss Anabaptists of 16th-Century <ENAMEX TYPE="LOCATION"> Europe </ENAMEX>
PERSON F Who is the Japanese Emperor? <ENAMEX TYPE="PER SON"> Akihito</ENAMEX>
MEASURE F How many people speak Gaelic in Scotland? <NUMEX TYPE="PERCENT"> 1.4 percent</NUMEX> of Scots

TIME F When did Latvia gain independence? <TIMEX TYPE="DATE"> 1991</TIMEX>
LOCATION F Where is the Al Aqsa Mosque? <ENAMEX TYPE="LOCA TION"> Jerusalem</ENAMEX>
OTHER F Name an animal that is capable of emitting light. glow-worms
PERSON F Who wrote "Ulysses"? <ENAMEX TYPE="PERSON"> James Joyce</ENAMEX>
LOCATION F Where is the Valley of the Kings? in <ENAMEX TYPE= "LOCATION"> Thebes</ENAMEX>
ORGANIZATION F What is the official German airline called? <ENA MEX TYPE="ORGANIZATION"> Lufthansa</ENAMEX>
TIME F When did Prince Charles and Diana get married? <TIMEX TYPE="DATE"> 1981</TIMEX>
MEASURE F How much did the Channel Tunnel cost? <NUMEX TYPE="MONEY"> $15-billion</NUMEX>
LOCATION F Where is Cedars-Sinai Medical Center? in <ENAMEX TYPE="LOCATION"> Los Angeles</ENAMEX>
TIME F In what year did Hurricane Andrew occur? <TIMEX TYPE= "DATE"> 1992</TIMEX>
ORGANIZATION D What is UEFA? European football's governing body
OTHER F Name a unit of radioactivity. curie
LOCATION F In what European country is Galway located? <ENAMEX TYPE="LOCATION"> Ireland</ENAMEX>
TIME F What year did the Olympic Games take place in Barcelona? <TIMEX TYPE="DATE"> 1992</TIMEX>
OTHER F What are breast implants made of? silicone-gel
PERSON F Who directed "Braveheart"? <ENAMEX TYPE="PERSON"> Mel Gibson</ENAMEX>
PERSON D Who is Yves Saint Laurent? fashion designer
TIME F When did Bosnia's secession from Yugoslavia occur? in <TIMEX TYPE="DATE"> early 1992</TIMEX>
OTHER F Name a French newspaper. Le Monde
TIME F When did the bomb attack at the World Trade Center occur? <TIMEX TYPE="DATE"> two years ago</TIMEX>
TIME F When was John Lennon killed? <TIMEX TYPE="DATE"> 1980</TIMEX>
OTHER F What do antioxidants do? deactivate free radicals
TIME F What year was Halley's comet visible? <TIMEX TYPE="DATE"> 1909</TIMEX>

TIME F On which day does the summer solstice fall? <TIMEX TYPE= "DATE"> June 21</TIMEX>
OTHER F What is salicylic acid extracted from? willow bark
LOCATION F Where is the Leaning Tower? <ENAMEX TYPE="LO CATION"> Pisa</ENAMEX>
PERSON D Who is Paul Simon? politician
ORGANIZATION D Who are the carabinieri? the paramilitary police corps that patrols <ENAMEX TYPE="LOCATION"> Italy</ENAMEX> 's 5,000 miles of seacoast
LOCATION F Where is Heathrow airport? <ENAMEX TYPE="LOCA TION"> London</ENAMEX>
PERSON F Who stars in "Sleepless in Seattle"? <ENAMEX TYPE="PER SON"> Tom Hanks</ENAMEX>
ORGANIZATION D What is the yakuza? the Japanese mafia
OTHER F What color is carbon monoxide? colorless
PERSON F Who is the Mayor of New York? <ENAMEX TYPE="PER SON"> Rudolph Giuliani</ENAMEX>
ORGANIZATION D What is the FDA? <ENAMEX TYPE="ORGANI ZATION"> Food and Drug Administration</ENAMEX>
LOCATION F Where is the Hermitage Museum? in <ENAMEX TYPE= "LOCATION"> St Petersburg</ENAMEX>
MEASURE F How many cardinals are entitled to elect the Pope? 120
PERSON F What is Armani's first name? <ENAMEX TYPE="PERSON"> Giorgio</ENAMEX>
OTHER F What nationality was Aristotle Onassis? Greek
LOCATION F What galaxy does Earth belong to? <ENAMEX TYPE= "LOCATION"> Milky Way</ENAMEX>
TIME F In what year was the Statue of Liberty built? <TIMEX TYPE= "DATE"> 1886</TIMEX>
OTHER F What are fiber-optic cables made up of? thin strands of glass
PERSON F Who painted the Guernica? <ENAMEX TYPE="PERSON"> Picasso</ENAMEX>
MEASURE F What is the population of Chechnya? 1.2 million people
OTHER F What type of government does France have? republic
MEASURE F How many continents are there? five
PERSON F What is the name of Kurt Cobain's wife? <ENAMEX TYPE="PERSON"> Courtney Love</ENAMEX>
OTHER F Name a board game. Scrabble
TIME F In what year did the Yom Kippur War take place? <TIMEX TYPE="DATE"> 1973</TIMEX>

MEASURE F How many people are killed by landmines every year? 10,000
LOCATION F What is the highest active volcano in Europe? <ENAMEX TYPE="LOCATION"> Mount Etna</ENAMEX>
LOCATION F What river flows through Dublin? <ENAMEX TYPE= "LOCATION"> Liffey</ENAMEX>
MEASURE F How many Bavarians are Catholic? Some <NUMEX TYPE="PERCENT"> 90 percent</NUMEX>
ORGANIZATION D What is Amnesty International? human rights group
PERSON D Who is Silvio Berlusconi? a conservative media magnate
OTHER F What does "Forza Italia" mean? Go, <ENAMEX TYPE="LO CATION"> Italy</ENAMEX> !
OTHER F What did the artist Christo wrap up? a <ENAMEX TYPE= "LOCATION"> Paris</ENAMEX> bridge
OTHER F What does a luthier make? guitar
OTHER F Tell me a reason for teenage suicides. a poor exam result
TIME F When was the Ulysses spacecraft launched? <TIMEX TYPE="DA TE"> 1990</TIMEX>
PERSON D Who are the Simpsons? a family who love each other and drive each other crazy
ORGANIZATION D What is freemasonry? a secret society
OTHER F What was Aldrich H. Ames accused of? spying for the Russians
LOCATION F Where is Red Square located? <ENAMEX TYPE="LOCA TION"> Moscow</ENAMEX>
OTHER F What band contributed to the soundtrack of the film "Zabriskie Point"? <ENAMEX TYPE="ORGANIZATION"> Pink Floyd</ENA MEX>
PERSON F Name a famous person who was photographed by Man Ray. <ENAMEX TYPE="PERSON"> James Joyce</ENAMEX>
TIME F What year did Pope John Paul II become pontiff? <TIMEX TYPE="DATE"> 1978</TIMEX>
OTHER F Who freed the town of Sainte-Mere-Eglise on D-day? <ENA MEX TYPE="LOCATION"> U.S.</ENAMEX> paratroopers
PERSON D Who is Alan Turing? the British scientist whom the authors rediscover as the uncrowned king of modern artifical intelligence
MEASURE F How old is Beck Hansen? 23
PERSON D Who is Harlequin? Italian commedia dell'arte figure
OTHER F In what war did the International Brigades fight? Spanish War

PERSON F Who was Russia's last czar? <ENAMEX TYPE="PERSON"> Nicholas II</ENAMEX>

OTHER F Tell me the name of a robot. Robodoc

OTHER F What animal coos? pigeon

MEASURE F How many pandas are there in the wild in China? fewer than 1,000

PERSON F Who plays the role of a prostitute in "Taxi Driver"? <ENAMEX TYPE="PERSON"> Jodie Foster</ENAMEX>

OTHER F What is acetic anhydride used for? to transform opium into heroin

PERSON F Who coined the expression "close encounters of the third kind"? Dr <ENAMEX TYPE="PERSON"> J Allen Hynek</ENAMEX>

LOCATION F What country does the tango come from? <ENAMEX TYPE="LOCATION"> Argentina</ENAMEX>

LOCATION F Name a country that exports rice. <ENAMEX TYPE="LOCATION"> Vietnam</ENAMEX>

LOCATION F What country is the main producer of diamonds? <ENAMEX TYPE="LOCATION"> South Africa</ENAMEX>

MEASURE F How many people were declared missing in the Philippines after the typhoon "Angela"? 280

MEASURE F How many astronauts were aboard the space shuttle Atlantis? eight

LOCATION F Where is the Reichstag? <ENAMEX TYPE="LOCATION"> Berlin</ENAMEX>

OTHER F What United States space shuttle took a Russian astronaut on board for the first time? Discovery

LOCATION F Where did the 1992 Olympic Games take place? <ENAMEX TYPE="LOCATION"> Barcelona, Spain</ENAMEX>

LOCATION F What continent is the ozone hole above? <ENAMEX TYPE="LOCATION"> Antarctica</ENAMEX>

PERSON F Who is the managing director of the International Monetary Fund? <ENAMEX TYPE="PERSON"> Michel Camdessus</ENAMEX>

ORGANIZATION F Name an oil company. <ENAMEX TYPE="ORGANIZATION"> Conoco</ENAMEX>

ORGANIZATION F Name a fast food chain. <ENAMEX TYPE="ORGANIZATION"> Burger King</ENAMEX>

ORGANIZATION F What party won the first multi-racial elections of South Africa? <ENAMEX TYPE="ORGANIZATION"> ANC</ENAMEX>

PERSON F Who won the Nobel Prize for Literature in 1994? <ENAMEX TYPE="PERSON"> Kenzaburo Oe</ENAMEX>
LOCATION F What is the capital of Venezuela? <ENAMEX TYPE="LOCATION"> Caracas</ENAMEX>
ORGANIZATION D What is the UNHCR? <ENAMEX TYPE="ORGANIZATION"> UN High Commissioner for Refugees</ENAMEX>
PERSON F Who is the German Minister for Economic Affairs? <ENAMEX TYPE="PERSON"> Guenter Rexrodt</ENAMEX>
MEASURE F How many people live in Bombay? 12 million
MEASURE F How many people live in Brazil? 152 million
MEASURE F How many inhabitants does South Africa have? nearly 43 million
ORGANIZATION F For what basketball team does Shaquille O'Neal play? <ENAMEX TYPE="ORGANIZATION"> Orlando Magic</ENAMEX>
TIME F In what year was the Chilean president Allende assassinated? <TIMEX TYPE="DATE"> 1973</TIMEX>
PERSON D Who is Juan Antonio Samaranch? <ENAMEX TYPE="ORGANIZATION"> IOC</ENAMEX> president
TIME F When did Pinochet come to power in Chile? <TIMEX TYPE="DATE"> 1973</TIMEX>
PERSON F What is the name of the chairman of the Federal Reserve Board? <ENAMEX TYPE="PERSON"> Alan Greenspan</ENAMEX>
TIME F What year was the NATO founded? <TIMEX TYPE="DATE"> 1949</TIMEX>
ORGANIZATION F What does the abbreviation OAU stand for? <ENAMEX TYPE="ORGANIZATION"> Organisation of African Unity</ENAMEX>
LOCATION F Where is Hyde Park? <ENAMEX TYPE="LOCATION"> N.Y.</ENAMEX>
ORGANIZATION F Of what political party is Ian Paisley the leader? Democratic Unionist
ORGANIZATION F What is the name of the national Belgian airline? <ENAMEX TYPE="ORGANIZATION"> Sabena</ENAMEX>
ORGANIZATION F Which company has its headquarters in Armonk? <ENAMEX TYPE="ORGANIZATION"> IBM</ENAMEX>
ORGANIZATION D What is UNICE? the <ENAMEX TYPE="ORGANIZATION"> European Employers' Federation</ENAMEX>
LOCATION F Where is CERN? in <ENAMEX TYPE="LOCATION"> Geneva</ENAMEX>

MEASURE F How many member states does CERN have? 19
MEASURE F How many inhabitants does Slovenia have? 2 million
OTHER F What does the company Victorinox produce? the Original <ENAMEX TYPE="ORGANIZATION"> Swiss Army</ENAMEX> Knife
OTHER F What is the world's highest mountain? <ENAMEX TYPE="LOCATION"> Everest</ENAMEX>
OTHER F What does Oracle sell? software and systems services for British Telecommunications' planned delivery of interactive multimedia services to homes
ORGANIZATION D What is the WTO? <ENAMEX TYPE="ORGANIZATION"> World Trade Organization</ENAMEX>
LOCATION F Where is Halifax located? <ENAMEX TYPE="LOCATION"> Canada</ENAMEX>
PERSON F Who is the Russian Minister of Finance? <ENAMEX TYPE="PERSON"> Andrei P. Vavilov</ENAMEX>
TIME F When did the attack at the Saint-Michel underground station in Paris occur? on <TIMEX TYPE="DATE"> July 25</TIMEX>
TIME F When did Lenin die? <TIMEX TYPE="DATE"> 1924</TIMEX>
TIME F When did the Iranian islamic revolution take place? <TIMEX TYPE="DATE"> 1979</TIMEX>
ORGANIZATION F Who committed the terrorist attack in the Tokyo underground? <ENAMEX TYPE="ORGANIZATION"> Aum doomsday cult/ENAMEX>
LOCATION F Where are UNESCO's headquarters? <ENAMEX TYPE="LOCATION"> Paris</ENAMEX>
ORGANIZATION F What is the name of Silvio Berlusconi's party? <ENAMEX TYPE="ORGANIZATION"> Forza Italia</ENAMEX>
TIME F When did Pearl Harbor's attack take place? <TIMEX TYPE="DATE"> Dec. 7, 1941</TIMEX>
PERSON F Who directed "Nikita"? <ENAMEX TYPE="PERSON"> Luc Besson</ENAMEX>
ORGANIZATION D What is the GIA? <ENAMEX TYPE="ORGANIZATION"> Armed Islamic Group</ENAMEX>
ORGANIZATION F What is Charles Millon's political party? <ENAMEX TYPE="ORGANIZATION"> Union for French Democracy</ENAMEX> (<ENAMEX TYPE="ORGANIZATION"> UDF</ENAMEX>)
PERSON F Who is the new president of Rwanda? Pasteur <ENAMEX TYPE="PERSON"> Bizimungu</ENAMEX>
ORGANIZATION F What does "UAE" stand for? <ENAMEX TYPE=

"LOCATION"> United Arab Emirates</ENAMEX>
ORGANIZATION F Of what organisation was Pierre-Paul Schweitzer general manager? <ENAMEX TYPE="ORGANIZATION"> International Monetary Fund</ENAMEX>
ORGANIZATION F What racing team is Flavio Briatore the manager of? <ENAMEX TYPE="ORGANIZATION"> Benetton</ENAMEX>
TIME F Since when has Iraq been under embargo? its <TIMEX TYPE= "DATE"> August 1990</TIMEX> invasion of <ENAMEX TYPE="LOCA TION"> Kuwait</ENAMEX>
PERSON F Who wrote "The Little Prince"? <ENAMEX TYPE="PER SON"> Antoine de Saint-Exupery</ENAMEX>
PERSON F Who did Whoopi Goldberg marry? <ENAMEX TYPE="PER SON"> Lyle Trachtenberg</ENAMEX>
PERSON F What is the name of the Queen of Denmark? <ENAMEX TYPE="PERSON"> Margrethe II</ENAMEX>
ORGANIZATION F What group killed Aldo Moro? <ENAMEX TYPE="ORGANIZATION"> Red Brigades</ENAMEX> guerrillas
ORGANIZATION F Of what team is Bobby Robson coach? <ENAMEX TYPE="ORGANIZATION"> Porto</ENAMEX>
MEASURE F How fast does light travel? 300,000km a second
TIME F When did Simon Bolivar die? <TIMEX TYPE="DATE"> 1830</TIMEX>
OTHER F What did the Titanic hit? an iceberg
ORGANIZATION F What country is the world football champion? <ENAMEX TYPE="LOCATION"> Brazil</ENAMEX>
OTHER F What does Faust sell to the devil? his soul
PERSON D Who is Jorge Amado? Brazilian novelist
OTHER F Name an odourless and tasteless liquid. NIL
LOCATION F How high above ground level is the ozone layer? NIL
OTHER F What is the former Argentinian currency? NIL
ORGANIZATION F Who manufactures Invirase? NIL
LOCATION F Where is the registered office of the European Monetary Institute? NIL
TIME F When was CERN founded? NIL
PERSON F Who is the director of CERN? NIL
TIME F When will Weimar be the European Capital of Culture? NIL
ORGANIZATION F Who produces the Smart compact car? NIL
TIME F When will the Guggenheim Museum in Bilbao be inaugurated? NIL

PERSON F What is the name of the Ukraine Prime Minister appointed in June of 1994? NIL
OTHER F What is the name of the only independent daily newspaper of Yugoslavia? NIL
PERSON F Who directed the film "Lisbon Story"? NIL
ORGANIZATION F Of what band is Teresa Salgueiro the vocalist? NIL
MEASURE F How many submarines has the Portuguese Navy? NIL

TIETOJENKÄSITTELYTIETEEN LAITOS
PL 68 (Gustaf Hällströmin katu 2 b)
00014 Helsingin yliopisto

DEPARTMENT OF COMPUTER SCIENCE
P.O. Box 68 (Gustaf Hällströmin katu 2 b)
FIN-00014 University of Helsinki, FINLAND

JULKAISUSARJA **A**

SERIES OF PUBLICATIONS **A**

Reports may be ordered from: Kumpula Science Library, P.O. Box 64, FIN-00014 University of Helsinki, FINLAND.

A-1998-1 G. Lindén & M. Tienari (eds.): Computer Science at the University of Helsinki 1998. 112 pp.

A-1998-2 L. Kutvonen: Trading services in open distributed environments. 231 + 6 pp. (Ph.D. thesis).

A-1998-3 E. Sutinen: Approximate pattern matching with the q-gram family. 116 pp. (Ph.D. thesis).

A-1999-1 M. Klemettinen: A knowledge discovery methodology for telecommunication network alarm databases. 137 pp. (Ph.D. thesis).

A-1999-2 J. Puustjärvi: Transactional workflows. 104 pp. (Ph.D. thesis).

A-1999-3 G. Lindén & E. Ukkonen (eds.): Department of Computer Science: annual report 1998. 55 pp.

A-1999-4 J. Kärkkäinen: Repetition-based text indexes. 106 pp. (Ph.D. thesis).

A-2000-1 P. Moen: Attribute, event sequence, and event type similarity notions for data mining. 190+9 pp. (Ph.D. thesis).

A-2000-2 B. Heikkinen: Generalization of document structures and document assembly. 179 pp. (Ph.D. thesis).

A-2000-3 P. Kähkipuro: Performance modeling framework for CORBA based distributed systems. 151+15 pp. (Ph.D. thesis).

A-2000-4 K. Lemström: String matching techniques for music retrieval. 56+56 pp. (Ph.D.Thesis).

A-2000-5 T. Karvi: Partially defined Lotos specifications and their refinement relations. 157 pp. (Ph.D.Thesis).

A-2001-1 J. Rousu: Efficient range partitioning in classification learning. 68+74 pp. (Ph.D. thesis)

A-2001-2 M. Salmenkivi: Computational methods for intensity models. 145 pp. (Ph.D. thesis)

A-2001-3 K. Fredriksson: Rotation invariant template matching. 138 pp. (Ph.D. thesis)

A-2002-1 A.-P. Tuovinen: Object-oriented engineering of visual languages. 185 pp. (Ph.D. thesis)

A-2002-2 V. Ollikainen: Simulation techniques for disease gene localization in isolated populations. 149+5 pp. (Ph.D. thesis)

A-2002-3 J. Vilo: Discovery from biosequences. 149 pp. (Ph.D. thesis)

A-2003-1 J. Lindström: Optimistic concurrency control methods for real-time database systems. 111 pp. (Ph.D. thesis)

A-2003-2 H. Helin: Supporting nomadic agent-based applications in the FIPA agent architecture. 200+17 pp. (Ph.D. thesis)

A-2003-3 S. Campadello: Middleware infrastructure for distributed mobile applications. 164 pp. (Ph.D. thesis)

A-2003-4 J. Taina: Design and analysis of a distributed database architecture for IN/GSM data. 130 pp. (Ph.D. thesis)

A-2003-5 J. Kurhila: Considering individual differences in computer-supported special and elementary education. 135 pp. (Ph.D. thesis)

A-2003-6 V. Mäkinen: Parameterized approximate string matching and local-similarity-based point-pattern matching. 144 pp. (Ph.D. thesis)

A-2003-7 M. Luukkainen: A process algebraic reduction strategy for automata theoretic verification of untimed and timed concurrent systems. 141 pp. (Ph.D. thesis)

A-2003-8 J. Manner: Provision of quality of service in IP-based mobile access networks. 191 pp. (Ph.D. thesis)

A-2004-1 M. Koivisto: Sum-product algorithms for the analysis of genetic risks. 155 pp. (Ph.D. thesis)

A-2004-2 A. Gurtov: Efficient data transport in wireless overlay networks. [B 141 pp. (Ph.D. thesis)

A-2004-3 K. Vasko: Computational methods and models for paleoecology. 176 pp. (Ph.D. thesis)

A-2004-4 P. Sevon: Algorithms for Association-Based Gene Mapping. 101 pp. (Ph.D. thesis)

A-2004-5 J. Viljamaa: Applying Formal Concept Analysis to Extract Framework Reuse Interface Specifications from Source Code. 206 pp. (Ph.D. thesis)

A-2004-6 J. Ravantti: Computational Methods for Reconstructing Macromolecular Complexes from Cryo-Electron Microscopy Images. 100 pp. (Ph.D. thesis)

A-2004-7 M. Kääriäinen: Learning Small Trees and Graphs that Generalize. 45+49 pp. (Ph.D. thesis)

A-2004-8 T. Kivioja: Computational Tools for a Novel Transcriptional Profiling Method. 98 pp. (Ph.D. thesis)

A-2004-9 H. Tamm: On Minimality and Size Reduction of One-Tape and Multitape Finite Automata. 80 pp. (Ph.D. thesis)

A-2005-1 T. Mielikäinen: Summarization Techniques for Pattern Collections in Data Mining. 201 pp. (Ph.D. thesis)

A-2005-2 A. Doucet: Advanced Document Description, a Sequential Approach. 161 pp. (Ph.D. thesis)

A-2006-1 A. Viljamaa: Specifying Reuse Interfaces for Task-Oriented Framework Specialization. 285 pp. (Ph.D. thesis)

A-2006-2 S. Tarkoma: Efficient Content-based Routing, Mobility-aware Topologies, and Temporal Subspace Matching. 198 pp. (Ph.D. thesis)

A-2006-3 M. Lehtonen: Indexing Heterogeneous XML for Full-Text Search. 185+3pp.(Ph.D. thesis).

A-2006-4 A. Rantanen: Algorithms for ^{13}C Metabolic Flux Analysis. 92+73pp.(Ph.D. thesis).

A-2006-5 E. Terzi: Problems and Algorithms for Sequence Segmentations. 141 pp. (Ph.D. Thesis).

A-2007-1 P. Sarolahti: TCP Performance in Heterogeneous Wireless Networks.(Ph.D. Thesis).

A-2007-2 M. Raento: TCP Exploring privacy for ubiquitous computing: Tools, methods and experiments. (Ph.D. thesis).