Linköping Studies in Science and Technology

Thesis No. 1320

Question Classification in Question Answering Systems

by

Håkan Sundblad

Submitted to Linköping Institute of Technology at Linköping University in partial fulfilment of the requirements for the degree of Licentiate of Philosophy

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2007

Question Classification in Question Answering Systems

by

Håkan Sundblad

June 2007
ISBN 978-91-85831-55-5
Linköping Studies in Science and Technology
Thesis No. 1320
ISSN 0280-7971
LiU-Tek-Lic-2007:29

ABSTRACT

Question answering systems can be seen as the next step in information retrieval, allowing users to pose questions in natural language and receive succinct answers. In order for a question answering system as a whole to be successful, research has shown that the correct classification of questions with regards to the expected answer type is imperative. Question classification has two components: a taxonomy of answer types, and a machinery for making the classifications.

This thesis focuses on five different machine learning algorithms for the question classification task. The algorithms are k nearest neighbours, naïve bayes, decision tree learning, sparse network of winnows, and support vector machines. These algorithms have been applied to two different corpora, one of which has been used extensively in previous work and has been constructed for a specific agenda. The other corpus is drawn from a set of users' questions posed to a running online system. The results showed that the performance of the algorithms on the different corpora differs both in absolute terms, as well as with regards to the relative ranking of them. On the novel corpus, naïve bayes, decision tree learning, and support vector machines perform on par with each other, while on the biased corpus there is a clear difference between them, with support vector machines being the best and naïve bayes being the worst.

The thesis also presents an analysis of questions that are problematic for all learning algorithms. The errors can roughly be divided as due to categories with few members, variations in question formulation, the actual usage of the taxonomy, keyword errors, and spelling errors. A large portion of the errors were also hard to explain.

This work has been supported by GSLT.

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Acknowledgements

This is where I thank all those who, in one way or the other, has made this journey a more pleasant one than it would have been without them. I prefer to do this in my mother tongue, which is why the remainder of this section will be in Swedish.

Först och främst vill jag tacka de som gjort det till sin livsuppgift att få detta arbete att nå sitt yttersta potential, nämligen min huvudhandledare **Arne Jönsson** och min bihandledare **Magnus Merkel**. Utan denna datorlingvistikens dynamiska duos insatser hade nog denna pamflett aldrig blivit av.

Jag måste även ta tillfället i akt att tacka övriga inom **HCS** i allmänhet och **NLPLAB** i synnerhet för att de utgjort en mycket trevlig och inspirerande forsknings- och fikamiljö.

Jag vill också tacka alla inom **GSLT**; doktorander, lärare, administratörer och annat löst folk.

Sist, men på intet sätt minst, vill jag tacka **Jenny**, **David** och **Ellen**.

Contents

1 Introduction

In Douglas Adams' novel *The Hitchhiker's Guide to the Galaxy* the computer Deep Thought is set to answer the Big Question about Life, the Universe and Everything. After seven and a half million years of computing it delivers the answer. Quite surprisingly, and much to the frustration of its creators, the answer it comes up with is forty-two. Part of the frustration stems from the fact that the semantics of the answer presented mismatches what would be expected from the question[1]. Incidentally, this thesis deals with questions, what kind of answers that can be expected from them, and how to automatically make those expectations.

1.1 Question answering

The goal of question answering (QA) is to provide a succinct answer, given a question posed in natural language (Hirschman & Gaizauskas 2001). We will here present a brief history of the field in question, present a more detailed definition of what a question answering really is, or can be, and also present the topic of this thesis, namely question classification.

[1] Admittedly, it is unclear what semantic class the answer to this question should belong to.

1.1.1 Brief history of the field

Although natural language question answering has seen a dramatic surge in interest since the turn of the millennium, it is by no means a new field in computer and information science. The earliest system was developed in 1959 (in the spirit of the era called *The Conversation Machine*), and by mid 1960's no less than fifteen different systems had been developed (Simmons 1965). One of the most memorable systems from the era was BASEBALL (Green et al. 1961). Although, capable of answering rather complex questions, BASEBALL was, not surprisingly, restricted to questions about baseball facts, and most question answering systems were for a long time restricted to front-ends to structured databases (cf. Androutsopoulos et al. 1995). However, since the introduction of the TExt Retrieval Conferences (TREC) question answering track there has been great progress in open-domain question answering (Voorhees 2001). These systems use unrestricted text as a primary source of knowledge.

1.1.2 Dimensions of the question answering

The problem of question answering can be described according to a number of different dimensions, each making the problem more or less complex (Hirschman & Gaizauskas 2001, Harabagiu et al. 2003, Carbonell et al. 2000, Moldovan et al. 2002). These dimensions can roughly be divided into: level of understanding and reasoning, type of data used as knowledge source and the actual knowledge sources used for answering questions, and whether the system is domain specific or domain independent.

Level of understanding Systems can be distinguished by their overall purpose, or level of "understanding". In this respect they fall into either the category of information seeking systems or reading comprehension systems. TREC focuses solely on the former category, which can be seen as extensions of traditional search engines. Reading comprehension systems, on the other hand, attempts to answer questions (who, what, when, where, and why) related to that specific text (Hirschman et al. 1999).

Type of data Another dimension regards the type of data from which answers are retrieved. Here we can distinguish between structured data (e.g. databases), semi-structured data (e.g. comment fields in databases), and free text (e.g. news wire collections). Question answering systems are nowadays almost exclusively concerned with free text.

Knowledge source Different question answering systems utilize different knowledge sources. These can broadly be divided into a single text, a single collection or book (e.g. an encyclopedia), a fixed set of collections (e.g. a news corpus drawn from different news wires), or the web. As mentioned above, reading comprehension systems are focused on a single text. TREC focuses on a fixed collection of news articles, but many systems participating in the contest are also capable of using the entire web as a knowledge source.

Generality We can also make a distinction between systems that are either domain specific or domain independent (open-domain). Domain specific systems attempt to answer questions within a single, more or less well-defined, domain (like BASEBALL described in section 1.1.1). TREC is concerned with open-domain systems, i.e., systems that attempts to answer whichever question a user wants to know the answer to.

1.1.3 The anatomy of question answering systems

The systems participating in the TREC QA track share quite a number of features and technologies, and the overall design of the systems are in most cases strikingly similar. Most systems treat question answering as three distinct sub-tasks: question processing, document processing, and answer processing (cf. Harabagiu et al. 2003). An illustration of this prototypical system architecture is shown in figure 1.1.

Question processing Question processing most often consists of construction of question representation, derivation of expected answer type, and keyword extraction. Some systems also perform question reformulation, where a question is transformed into a number of declarative equivalents.

Parsing is done in order to construct some form of structural representation of the question. Typically, the structure is a syntactic tree or a dependency tree. This structure can subsequently be used to locate and verify answers within retrieved documents and passages (Paşca 2003). The extracted keywords are used by retrieval engines to fetch relevant documents. The expected answer type is also derived for this purpose.

Document processing Document processing typically includes keyword expansion, document retrieval, and passage identification. Keyword expansion typically involves taking the keywords extracted in the question

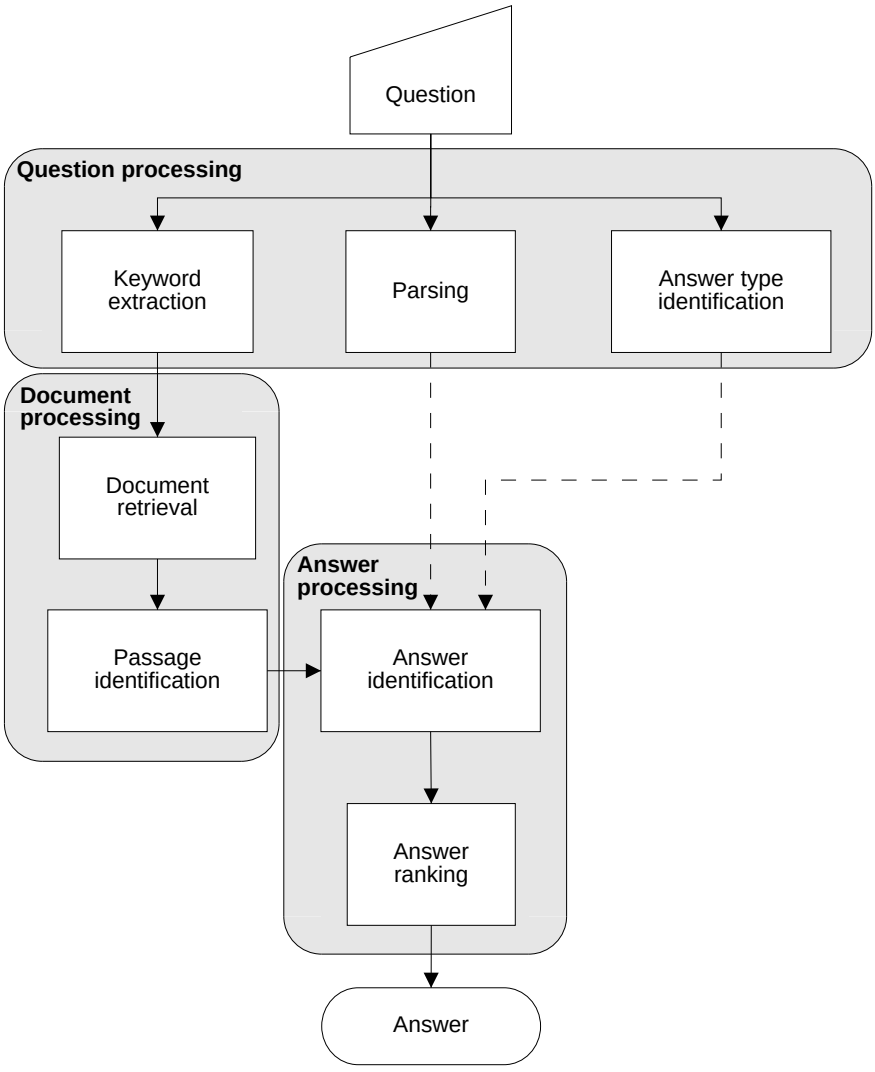

Figure 1.1: Prototypical serial question answering system architecture.

processing stage and looking them up in a thesaurus, or other resource, and adding similar search terms in order to fetch as many relevant documents as possible. A term such as "kill" might be expanded to "murder" and "assassinate" for instance. Document retrieval is in essence restricted to passing the expanded keywords to a standard search engine (e.g. Google) and retrieving the documents with highest ranks. In passage retrieval, within

each document the paragraph or section containing the possible answer is identified.

Answer processing Answer processing can consist of candidate answer identification, answer ranking, and answer formulation. Identifying the candidate answers means taking the results from the passage identification phase and further processing it. Often this means doing a full parse of the passage and comparing it to a full parse of the question. This results in a set of candidate answers that are then ranked according to an algorithm or set of heuristics. Answer formulation is in most cases skipped completely and the answer is presented as it was found in the document.

1.1.4 Question classification

As mentioned, one part of the question processing stage is question classification. The derivation of expected answer types is often carried out by means of machine learning approaches. This task relies on three parts: a taxonomy of answer types into which questions are to be classified, a corpus of questions prepared with the correct answer type classification, and an algorithm that learns to make the actual predictions given this corpus. For this to be successful, the taxonomy needs to be well designed and the corpus must be correct and of appropriate size. The correct prediction of the answer type has in fact been shown to be one of the most important factors for a question answering system to succeed is the ability to correctly identify the expected answer's semantic type. The following quote is from Moldovan et al. (2002):

> "Whenever the derivation of the expected answer type [. . .] fails, the set of candidate answers identified in the retrieved passages is either empty in 28.2% of the cases (when the answer type is unknown) or contains the wrong entities for 8.2% (when the answer type is incorrect)."

If question classification is successful, the system might even use different processing strategies (cf. Harabagiu et al. 2001, Kazawa et al. 2001, Prager et al. 2003) to answer different types of questions. Throughout this work question classification should be taken as synonymous to this specific notion, i.e. the prediction of an expected answer type. Question classification, question categorization, and answer type recognition are used interchangeably.

1.2 Research issues and motivations

The focus of this thesis is question classification. More specific, classification of questions with respect to expected answer types. The research is to a large extent motivated by the fact that previous research has shown that correctly predicting the expected answer type is imperative for a question answering system as a whole to be successful (Moldovan et al. 2002). Work on this has been done in the field, but due to the fact that most of this research has been done in more or less direct connection to TREC, the results might in some ways be biased. The reason for this bias is that TREC has an explicit agenda that changes from year to year, and the material used therefore is designed to fit a specific purpose. More specifically, the corpora provided are to some extent actual users' questions, but manual selection has been done, as well as the addition of "interesting" questions intended to test the limits of the participating systems. The first research issue is therefore:

Research issue 1 *Is the corpus used in much of the previous work on question classification biased, and if so what can be expected in terms of performance on a set of actual users questions?*

In order to establish this we will run five of the most commonly used machine learning algorithms (Naïve Bayes, *k* Nearest Neighbors, Decision Tree Learning, Sparse Network of Winnows, and Support Vector Machines) in previous work on the TREC material in order to establish their performance on this corpus, and then re-run the same algorithms on another corpus derived from logs of actual users questions to a running on-line system in order to compare the results.

A second aim of the thesis is to make an analysis of the performance of machine learning algorithms. More specifically, we will look at the cases were they fail and attempt to establish whether the failure is due to inherent properties of questions per se or the taxonomy into which the algorithms are to classify the questions

Research issue 2 *Are there questions that are problematic for all algorithms, or are there differences between them? Are the problems due to the questions or the taxonomy into which they are categorized?*

This has not been done before in the literature. As a result of the work an evaluation of the most widely used taxonomy in the field has been conducted.

1.3 Contributions

The present work contributes the following:

- A baseline of the performance of standard machine learning techniques for the problem of question classification.
- A ranking of standard ML algorithms for the problem at hand based on significance testing.
- An analysis of questions that are difficult to classify.

1.4 Thesis outline

The outline of the present thesis is as follows:

Introduction The current chapter where we have introduced the field, established the research issues, and outlined the contributions.

Background This chapter contains a formal definition of question classification, as well as a review of the techniques and tools that have been used in previous work to attack the problem.

Method This chapter contains a detailed description of the methodology used to answer the research issues. It details which algorithms are used and under which premises, the corpora and taxonomy used, and the type of significance testing that is performed.

Taxonomy This chapter contains a detailed analysis of the taxonomy used throughout the work. The taxonomy is presented in detail, and is then scrutinized in order to establish the pros and cons of using that taxonomy.

Experiments This chapter presents the results from the experiments performed on the machine learning algorithms on the question classification problem. The chapter also contains a qualitative analysis of the most problematic questions.

Conclusions and future work This chapter contains a discussion of the results from the experiments, and also presents the conclusions that can be drawn from them. It concludes with possible directions for future work.

2
Question classification

This chapter starts of by defining the problem of question classification, both informally and formally. It then proceeds to review different taxonomies of answer types that have been used in previous work. The chapter is concluded with an overview of the different approaches taken so far to automatically classify questions with regards to the expected answer type.

Question classification can losely be defined as: given a question (represented by a set of features), assign the question to a single category or a set of categories (answer types). Before we review actual approaches to the problem, a formal definition is presented, as well as a section on how the performance is typically evaluated.

2.1 A definition of question classification

Adopting the formal definition of text categorization (Sebastiani 2002) to the problem of question classification, the task can be defined as follows:

Definition 1 *Question classification is the task of assigning a boolean value to each pair* $\langle q_j, c_i \rangle \in \mathcal{Q} \times \mathcal{C}$*, where* $\mathcal{Q}$ *is the domain of questions and* $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ *is a set of predefined categories.*

Assigning $\langle q_j, c_i \rangle$ to the value T indicates that q_j is judged to belong to the category c_i, while an assigment to the value F indicates that q_j is *not* judged as belonging to the category c_i.

In a machine learning setting, the task is to make the unknown target function $\hat{\Phi} : \mathcal{Q} \times \mathcal{C} \rightarrow \{T, F\}$ approximate the ideal target function $\Phi : \mathcal{Q} \times \mathcal{C} \rightarrow \{T, F\}$, such that $\hat{\Phi}$ and Φ coincide as much as possible.

2.2 Answer type taxonomies

As we saw in the definition above, question classification involves two parts. First, the questions to be categorized per se, and second, a set of categories into which the questions are to be categorized. This set of categories is from now on referred to as a taxonomy of possible answer types. This section presents a brief review of different answer type taxonomies that have been used in question answering systems.

Answer type taxonomies can be divided into flat and hierarchical taxonomies[1]. Flat taxonomies have only one level of categories, and are therefore not taxonomies in the true sense of the word. Hierarchical taxonomies consists of super- and sub-categories. For example, if we had the two concepts vehicle and car, in a flat taxonomy both would be considered as being on the same level of granularity, while in a hierarchical taxonomy the latter would be considered a sub-category of the former.

Some systems utilize (e.g. Srihari & Li 1999) taxonomies derived from the categories used in the Message Understanding Conference (MUC) evaluations. MUC distinguished between named entities (organization, person, and location), temporal expressions (date and time), and number expressions (money and percentage) (Chinchor 1997).

A common solution to constructing answer type taxonomies is to manually extract a subset of WordNet (Fellbaum 1998). Kim et al. (2000) use this approach to construct a taxonomy of 46 different semantic different categories[2]. Harabagiu et al. (2000) has a taxonomy with a number of top categories that are connected (many-to-many) to several word classes in the WordNet database. Another approach is to manually analyse a specific corpus, i.e. a collection of texts or questions, and infer a taxonomy from it. This is the approach used by Li & Roth (2002).

[1] An overview of flat taxonomies can be found in appendix A, and a corresponding overview of hierarchical taxonomies is presented in appendix B.

[2] Kim et al. (2000) are omitted from the overviews in appendix A since they do not present exactly which categories they use.

Ogden et al. (1999) use the one layered answer type taxonomy described in table A.1. The concepts are drawn from the general, hierarchical Mikrokosmos Ontology (Mahesh & Nirenburg 1995).

Suzuki, Taira, Sasaki & Maeda (2003) use the taxonomy described in table B.1, which consists of a total of 150 different categories. The taxonomy is hierarchical, with a maximum depth of 5. The taxonomy is based on a corpus consisting of 5011 different questions in Japanese[3]. To find a given node's parent, trace backwards until a node one level up is found. No motivation for the taxonomy is given.

It is worth noticing that Suzuki, Taira, Sasaki & Maeda (2003) report that less than 1% of the questions were labelled as other, which would indicate that the taxonomy fits the corpus very well.

Li and Roth, 2002 Li & Roth (2002) define a two-layered taxonomy. This taxonomy is described in depth here since it is the taxonomy used in the rest of the work in the thesis.

The taxonomy consists of 6 coarse categories and a total of 50 finer categories and unfortunately there is no description of how the taxonomy was constructed or motivations for the specific categories are given. The full taxonomy is presented in table 2.1. The descriptions are those provided by Li & Roth (2002).

The coarse classes are as follows: abbreviation (abbr), entity (enty), description (desc), human (hum), location (loc), and numeric (num).

The abbreviation category consists of two subcategories. One subcategory concerns how acroyms should be exanded (abbr:exp) and the other concerns how to abbreviate a given term (abbr:abb).

The entity category handles questions that asks for a specific object that fits a description, e.g. "What are the languages spoken by the natives in Afghanistan?" (enty:lang) or "What does the Peugeot company manufacture?" (enty:product). There are 22 subcategories to this coarse class.

The description category concerns questions that asks for more elaborate answers. The category consists of the following subcategories: definition (desc:def), description (desc:desc), manner (desc:manner), and reason (desc:reason). Definitions basically refer to encyclopedic definitions and a typical question might be "What is an ecological niche?". The descrition category covers questions like "What happened to Pompeii?" that needs an elaborate factual description of a term or event. Manner refers to questions like "How does a bill become law?" that asks for a description of a

[3]The exact nature of these questions are unclear due to poor command of Japanese.

Class	Definition	Class	Definition
ABBREVIATION	abbreviation	HUMAN	human beings
abb	abbreviation	group	a group or organization of persons
exp	expression abbreviated	ind	an individual
ENTITY	entities	title	title of a person
animal	animals	description	description of a person
body	organs of body	LOCATION	locations
color	colors	city	cities
creative	inventions, books and other creative pieces	country	countries
currency	currency names	mountain	mountains
dis.med.	diseases and medicine	other	other locations
event	events	state	states
food	food	NUMERIC	numeric values
instrument	musical instrument	code	postcodes or other codes
lang	languages	count	number of sth.
letter	letters like a-z	date	dates
other	other entities	distance	linear measures
plant	plants	money	prices
product	products	order	ranks
religion	religions	other	other numbers
sport	sports	period	the lasting time of sth.
substance	elements and substances	percent	fractions
symbol	symbols and signs	speed	speed
technique	techniques and methods	temp	temperature
term	equivalent terms	size	size, area and volume
vehicle	vehicles	weight	weight
word	words with a special property		
DESCRIPTION	description and abstract concepts		
definition	definition of sth.		
description	description of sth.		
manner	manner of an action		
reason	reasons		

Table 2.1: Li and Roth's taxonomy.

process. Finally, the reason category covers all why-questions. These are perhaps the most difficult questions to answer. Sometimes, an answer can be looked up in an encyclopedia and sometimes there is no clear answer. One of the goals, or perhaps dreams, of question answering systems is for the system to be able to make inferences by itself from different sources of information and present the user with an answer as well as the inference chain.

The human category covers questions relating to specific humans or human organizations. The individual (hum:ind) covers questions that asks for a specific person that fits a given description, such as "Who invented the Moog Synthesizer?". The group category (hum:gr) has the same purpose, but concerns questions where the answer rather is a group or organizations of people, such as a company. There is also a title category (hum:title) for questions that asks for a persons profession or title. For questions that requests information about a person, such as Who is "Colin Powell?", there is a description category (hum:desc).

The locations class covers geographic and virtual locations. Four subcategories covers geographic locations: city (loc:city), country (loc:country), mountain (loc:mount), and state (loc:state). A fifth category (loc:other) covers other geographic (e.g. planets and rivers) and also virtual locations (e.g. web adresses).

Finally, the numeric coarse class covers questions that requests for some kind of numeric information, such as dates, prices, ages and speed. There are 13 subcategories, 12 concerning specific numeric information and one category for those that does not fit the other 12 (num:other).

In the paper by Li & Roth (2002), the distribution of 500 TREC 2001 questions over the different categories are presented. In the coarse class entity 13% of the questions are labelled as other, in numeric 11% are labelled as other, and in location 62% are labelled as other. This indicates that there might be a need for more finer categories at least within the location class.

As we will see, this taxonomy has been used in some of the other work on question classification in question answering systems, and has become a kind of informal standard.

2.3 Rule-based approaches to question classification

Given a taxonomy of answer types, we also need a machinery for making the actual classification of questions into that taxonomy. Two different approaches to this can be distinguished: *rule-based classification* and *machine-learned classification*. This section deals with the former, while the next section presents a thorough review of the latter.

The arguably most straightforward approach to question classification, and one that has been adopted by many (Breck et al. 1999, Eichmann & Srinivasan 1999, Ferret et al. 1999, Hull 1999, Humphreys et al. 1999, Moldovan et al. 1999, Oard et al. 1999, Ogden et al. 1999, Prager et al. 1999, Radev et al. 2002, Singhal et al. 1999), is to use some form of, more or less com-

plex, hand-written rules and heuristics. These range from using only the surface form of questions to using tagging, parsing and semantics.

Singhal et al. (1999) use rules operating on words, as well as shallow parse information:

- Queries starting with Who or Whom are taken to be of type person.
- Queries starting with Where, Whence, or Whither are taken to be of type location.
- Queries starting with How few, How great, How little, How many, or How much are taken to be of type quantity.
- Queries starting with Which or What, lookup head noun in lexicon to determine answer type.

Hermjakob (2001) also use hand-crafted rules, but these operate on questions parsed both with syntactic and semantic information. Hermjakob use a deterministic, machine-learning based shift-reduce parser, CONTEX (Hermjakob & Mooney 1997), which is trained on a tree-bank consisting of Wall Street Journal (WSJ) sentences, questions from TREC-8 and TREC-9, as well as questions from a travel guide phrase book and on-line resources. The taxonomy used consists of 122 answer types (see appendix B, Hovy et al. (2002) for details[4]), or Qtargets as Hermjakob refers to them. The Qtargets are based on the analysis of 18,000 on-line questions.

The 122 Qtargets are computed based on a list of 276 handwritten rules. The reason that 276 rules can categorize 122 different answer types are, according to the author, that given a semantic parse tree, the rules can be formulated on a high level of abstraction. With the CONTEX parser trained on 200 WSJ sentences and 975 questions from the various sources mentioned above the system achieved a 96.1% accuracy of Qtarget classification given 179 unseen sentences. When trained on only WSJ sentences, the system achieved 65.3% accuracy.

2.4 Machine-learning approaches to question classification

This section presents different machine learning approaches that have been used for question classification.

[4]Note that the number of answer types has increased since the paper was published and that is why there are more than a 122 answer types listed in the appendix.

2.4.1 Decision rule learning with set-valued features

Radev et al. (2002) experiments with machine learning for question classification using decision rule learning with set-valued features (Cohen 1996). This is a standard decision tree/rule approach (Mitchell 1997) that has been augmented in that instead of being restricted to features with single values, the values can also be a set of values. The answer type taxonomy consists of 17 types (see appendix A), and the training data is TREC-8[5] and TREC-9[6] data. Testing data is TREC-10[7]. In the experiment, questions are represented by 13 features, 9 of which are semantic features based on WordNet (Fellbaum 1998). The classifier reached an accuracy of around 70%.

2.4.2 Sparse Network of Winnows (SNoW)

Li & Roth (2002) use a Sparse Network of Winnows (SNoW) to classify questions with respect to their expected answer type. SNoW[8] (Roth 1998) is a general learning architecture framework designed for learning in the presence of a very large number of features and can be regarded as a general purpose multi-class classifier. The learning framework is a sparse network of linear functions over a predefined or incrementally acquired feature space.

Li & Roth (2002) use the taxonomy described in section 2.2 (see appendix B for a full specification). The corpus used consisted of 4,500 questions published by USC[9], about 500 manually constructed questions for rare cases, and 894 TREC-8 and TREC-9 questions. The authors somehow amounts this to a training corpus of 5,500 questions. 500 questions from TREC-10 were used as a test corpus. The questions were manually labeled according to the taxonomy with exactly one label.

The classifier is actually a sequence of two classifiers - a coarse classifier and fine classifier. The idea is that the first classifier categorize a given question into one of the 6 coarse classes, while the second classifies the question into one of the fine classes belonging to the identified coarse class. By having a coarse classification first, the set of possible class labels (the confusion set) can be reduced and this might simplify the classification.

[5] `http://trec.nist.gov/data/qa/T8_QAdata/topics.qa_questions.txt`

[6] `http://trec.nist.gov/data/qa/T9_QAdata/qa_questions_201-893`

[7] `http://trec.nist.gov/data/qa/2001_qadata/main_task_QAdata/qa_main.894-1393.txt`

[8] SNoW and all the training and testing data used by Li & Roth (2002) is available at `http://l2r.cs.uiuc.edu/~cogcomp/`.

[9] The authors cite (Hovy et al. 2001) for details, but the nature of the questions is not clear from that paper.

This was not the case, since a single multi-class classifier performed as well as the hierarchical classifier.

The input to the classifiers were a list of features. The features used were words, part-of-speech tags (SNoW-based tagger), chunks (Abney 1991), named entities, head chunks (e.g. the first noun chunk in a sentence), and semantically related words (words that often occur with a specific question class). The last feature, semantically related words, were not created fully automatically. Some human intervention and manual selection had to be performed. Apart from these primitive features, a set of operators were used to compose more complex features. The semantically related words were constructed semi-automatically. The total feature space were around 200,000 features. The performance of both the flat and hierarchical classifier are presented in table 2.2.

	Precision for features					
Classifier	Word	PoS	Chunk	NE	Head	RelWord
Flat	52.40	77.20	77.00	78.40	76.80	84.00
Hierarchical	77.60	78.20	77.40	78.80	78.80	84.20

Table 2.2: Results from Li & Roth (2002).

As can be seen in table 2.2 the differences in performance between the flat and hierarchical classifiers are marginal in all cases except when only the words were used. If we disregard the case where semantically related words were used, which is semi-automatic, the highest precision reached was 78.80%.

2.4.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) has gained much interest as a learning approach to question classification (Zhang & Lee 2003b, Suzuki, Taira, Sasaki & Maeda 2003, Hacioglu & Ward 2003). SVM is an approach to machine learning developed by V. N. Vapnik (Cortes & Vapnik 1995) that has proven very successful in text categorization research (Joachims 1998, Dumais et al. 1998). SVM:s are binary classifiers, where the idea is to find a decision surface (or a hyperplane if the training examples are linearly separable) that separates the positive and negative examples while maximizing the minimum margin. The margin is defined as the distance between the decision

surface and the nearest positive and negative training examples (called *support vectors*). At the heart of SVM:s is the so called kernel function and new kernel functions are proposed continuously. Although SVM:s are binare classifiers, they can be extended to solve multi-class classification problems, such as question classification.

Tree kernel

Zhang & Lee (2003b) performed a number of experiments on question classification using the same taxonomy as Li & Roth (2002) (see appendix B), as well as the same training and testing data (see section 2.4.2). In an initial experiment they compared different machine learning approaches with regards to the question classification problem: Nearest Neighbors (NN), Naïve Bayes (NB), Decision Trees (DT), SNoW, and SVM. NN, NB, and DT are by now fairly standard techniques and good descriptions of them can be found in for instance Mitchell (1997). The feature extracted and used as input to the machine learning algorithms in the initial experiment was bag-of-words and bag-of-*n*grams (all continuous word sequences in the question). Questions were represented as binary feature vectors since the term frequency of each word or *n*gram in a question usually is 0 or 1[10]. The results of the experiments are shown in table 2.3.

	bag-of-words		**bag-of-*n*grams**	
Algorithm	coarse	fine	coarse	fine
NN	75.6	68.4	79.8	68.6
NB	77.4	58.4	83.2	67.8
DT	84.2	77.0	84.2	77.0
SNoW	66.8	74.0	86.6	75.8
SVM	85.8	80.2	87.4	79.2

Table 2.3: Results from Zhang & Lee (2003b).

The results for the SVM algorithm presented in table 2.3 is when the linear kernel is used. This kernel had as good performance as the polynomial, RBF, and sigmoid kernels (Zhang & Lee 2003b).

[10]This is of course a simplification. In a question such as "What is the name of the author of 'The name of the rose"' the word 'the' occurs four times and 'name' occurs two times. But in general this has no effect on the machine learning.

In a second experiment the linear kernel of the SVM was replaced with a tree kernel developed by the authors (Zhang & Lee 2003b). Since both the bag-of-words and the bag-of-*n*grams approaches ignore the syntactic structure of questions, certain questions can not be discriminated. The gist of the tree kernel is that a question is initially parsed into a tree structure using a parser like Charniak's (2000). The question is then represented as a vector in a high dimensional space which is defined over tree fragments. The tree fragments of a syntactic tree are all its sub-trees which include at least one terminal symbol (word) or one production rule. A similar tree kernel has previously been proposed by Collins & Duffy (2001), but Zhang & Lee's (2003b) definition allows the tree kernel to back off to a word linear kernel, which guarantees a performance at least at par with such a kernel. Using the tree kernel, the SVM was trained for coarse classification on the same data as before, yielding a precision of 90.0%. The results using the tree kernel is significantly better than both the word linear kernel and the *n*gram linear kernel ($p < .005$ and $p < .025$ respectively). Under the fine-grained category definition, the SVM based on the tree kernel is reported to only make slight improvements.

One thing should be noted with regards to the results presented in Zhang & Lee (2003b). It seems that for SVM the parameters of the tree kernel have been optimized for the best performance, while the other algorithms have been used with default values.

For all experiments involving SVM the one-against-one strategy for multi-class classification was adopted (Hsu & Lin 2002) and the LIBSVM[11] (Chang & Lin 2001) implementation of the SVM algorithm was used.

Hierarchical directed acyclic graph (HDAG) kernel

Suzuki, Taira, Sasaki & Maeda (2003) used SVM with a hierarchical directed acyclic graph (HDAG) kernel (Suzuki, Hirao, Sasaki & Maeda 2003) for the question classification problem. The HDAG kernel is specifically designed to handle structured natural language data and can handle structures within texts (such as chunks) as the features of texts without converting the structures to the explicit representation of numerical feature vectors.

The answer type taxonomy used by Suzuki, Taira, Sasaki & Maeda (2003) consists of 150 different types (see section 2.2 above for details) and appendix B for details). The corpus used was in Japanese and consisted of

[11] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

1011 questions from NTCIR-QAC[12], 2000 questions of CRL-QA[13] data, and 2000 other questions reported to be of TREC-style (Suzuki et al. 2002). After removing answer types with too few (less than 10) examples, a total of 68 answer types were actually used.

Suzuki, Taira, Sasaki & Maeda (2003) compared the SVM with HDAG kernel (SVMHDAG) with the following approaches: bag-of-words with first degree polynomial kernel (SVMBoW1), bag-of-words with second degree polynomial kernel (SVMBoW2), and SNoW using only bag-of-words (SNoWBoW). Four feature sets were created for each method: words only (W), words and named entities (W+N), words and semantic information (W+S), and words, named entities, and semantic information (W+N+S). The precision of the different approaches are summarized in table 2.4.

	Precision for features			
Classifier	W	W+N	W+S	W+N+S
SVMHDAG	73.0	73.6	74.2	74.9
SVMBoW1	67.8	69.1	68.6	70.4
SVMBoW2	67.9	68.6	67.1	69.4
SNoWBoW	56.2	57.3	61.4	62.6

Table 2.4: Results from Suzuki et al. (2003).

The results in table 2.4 show that the SVM using the HDAG kernel has better performance using any of the feature sets than the other approaches. This could indicate that structural information has a great impact on precision.

For all experiments involving SVM, a hierarchical decision model consisting of one-against-all (Hsu & Lin 2002) classifiers was constructed in order to enable multi-class classification. The authors also note that the classification gets worse for every level of classifiers that is traversed, which might indicate that using a hierarchy of classifiers might be inappropriate for the problem at hand.

Error correcting codes

Hacioglu & Ward (2003) use SVM with error correcting codes to convert the

[12] `http://www.nlp.cs.ritsumei.ac.jp/qac`

[13] `http://www.cs.nyu.edu/~sekine/PROJECT/CRLQA`

multi-class classification problem into a number of binary ones. In essence each class is assigned a codeword of 1's and -1's of length m, where m equals or is greater than the number of classes. This splits the multi-class data into m binary class data. Therefore, m SVM classifiers can be designed and their output combined. The SVM:s also used linear kernels.

The same taxonomy, training and testing data was used as in Li & Roth (2002) (see section 2.4.2 and appendix B). The features used was the 2000 most informative terms (IT) and named entity information (NE). Two different NE settings were used, one were only 7 different NE:s were considered (NE-7) and one where 29 different NE:s were considered (NE-29). The results are shown in table 2.5.

Method	1-gram	2-gram	3-gram
IT	79.4	80.2	78.4
NE-7	81.4	82.0	80.2
NE-29	75.4	78.6	78.8

Table 2.5: Results from Hacioglu & Ward (2003).

The reason that the NE-29 setting performs worse than the NE-7 is that the named entity tagger performance was worse in the former case than the latter.

2.4.4 Language Modeling-based Classification (LMC)

Pinto et al. (2002) use statistical language modeling (cf. Rosenfeld 2000) for question classification[14]. Language modeling classification is a generalization of the Naïve Bayes algorithm, but allows longer contexts and can benefit from better smoothing techniques in the presence of sparse data.

A statistical language model, in the case of question classification, is simply a probability distribution $P(Q)$ over all possible questions Q. The goal is to find the class c to which a given question q belongs. Using a Bayes classifier, the solution $\hat{c}$ is:

[14] Zhang & Lee (2003a) also present experiments with language modeling for question classification. The work bears strikning resemblance with the work by Pinto et al. (2002). The only difference seem to be that the former use smoothed bigrams, the latter use unprocessed bigrams. However, Zhang & Lee (2003a) do not acknowledge the work by Pinto et al. (2002).

$$\hat{c} = \arg\max_c P(c|q) = \arg\max_c P(q|c) \cdot P(c) \tag{2.1}$$

The prior probability $P(c)$ can be estimated by the fraction of training questions labeled c. The probability $P(q|c)$ can in turn be estimated by a language model. Pinto et al. (2002) use both a unigram model

$$P(q|c) = P(w_1|c) \cdot P(w_2|c) \cdot \ldots \cdot P(w_n|c) \tag{2.2}$$

and a bigram model

$$P(q|c) = P(w_1|c) \cdot P(w_2|c, w_1) \cdot \ldots \cdot P(w_n|c, w_n) \tag{2.3}$$

The corpus used by Pinto et al. (2002) drawn from the GovBot logs[15], TREC questions, and some other questions of unspecified origin. The questions were generalized in that a question like "Who is the president of the US?" became "Who is the president of the location?". In an attempt to improve the models the quesíons were also generalized another step to contain part-of-speech tags, e.g., "Who is the NP of the location?". The answer type taxonomy consists of seven different categories and can be found in appendix A. The results of the experiment in terms of precision can be found in table 2.6.

	Tagging	
Model	No	Yes
Unigram	.74	.56
Bigram	.73	.60

Table 2.6: Results from Pinto et al. (2002).

There seem to be no difference between the performance of the unigram and bigram model with or without the use of tags. Pinto et al. (2002) also constructed a regular expression classifier with a precision of 59%. A hybrid classifier using both regular expressions and language modeling obtained a precision of 89% on the TREC questions.

[15]GovBot was a database of US government web sites available for public search

2.5 Summary

This chapter began by defining the problem of question classification formally. We saw that addressing the problem needs three things: a corpus of questions to be categorized, a taxonomy into which categorization is to be made, and a machinery for actually performing the categorization. We then looked at different kinds of taxonomies that have been used in previous work. We concluded by looking briefly into rule-based classification, and deeply into machine learning-based approaches. We saw that some machine-learning approaches seem to be quite good at classifying questions, achieving a precision of around 85%. The next chapter contains a detailed description of the research issues addressed in the thesis, as well as the methodology and tools used.

3 Method

The chapter begins by elaborating on the research issues described in chapter 1. In order to investigate these issues a corpus consisting solely of real users' questions is needed and how this corpus was constructed is described next. The chapter then proceeds to describe what machine learning algorithms that have been used in the experiments, as well as under what premises. The chapter concludes with details of how the results of the experiments were evaluated, what measures were used and what they actually measure.

3.1 Research issues

As stated in chapter 1 the purpose of this thesis is to answer the following questions:

1. Is the corpus used in previous work on question classification biased?

2. How well do current answer type taxonomies fit a random sample of user questions?

3. How well do current techniques for automatic recognition of answer types work given a random sample corpus?

4. Are some classes of questions more problematic than others, and if so, which?

In order to address these issues a series of experiments will be conducted. First we will establish a baseline to which we can compare our results. This baseline is essentially established by re-examining how standard machine learning techniques perform when presented with the corpus and taxonomy built by Li & Roth (2002). This is the corpus and taxonomy that has been used in much of the previous work in the field. However, since no ranking of the different machine learning techniques have been established by means of significance testing, this is the first goal of the present work.

We will then go on to build a new annotated corpus, based on the same taxonomy as defined by Li & Roth (2002). The aim of this corpus is to represent a fair sample of actual users questions, as they can be expected to be presented to fully functional question answering systems. Fortunately, such a corpus exists in the AnswerBus logs. However, this is an untagged corpus, and the methodology for tagging it is described below. The machine learners will then be tested on this new corpus, and the results will be compared to the results from the baseline experiments. Essentially, this will yield a more accurate baseline to which future experiments in the field can compare their results. For present purposes, five algorithms will be tested: Naïve Bayes, *k* Nearest Neighbors, Decision Tree Learning, Sparse Network of Winnows, and Support Vector Machines. These are the most commonly used in previous work in the field (see chapter 2).

As a bi-product, an informal evaluation of the taxonomy proposed by Li & Roth (2002) will be performed. This evaluation is considered a partial result of the thesis.

As a final step, a detailed study of exactly which kinds of questions are problematic for machine learning algorithms will be conducted. To our knowledge such a study has not been done in the field. More specific, we will try to establish if there are certain questions that are inherently difficult to classify, or if perhaps the taxonomy is the problem.

To sum up, the following work is the scope of the thesis:

1. Re-examine previous work using the taxonomy and corpus established by Li & Roth (2002) and the algorithms: Naïve Bayes, *k* Nearest Neighbors, Decision Tree Learning, Sparse Network of Winnows, and Support Vector Machines

2. Build a new corpus based on the AnswerBus logs using the same taxonomy (see section 3.2)

3. Run the classifiers on this new corpus. The results of these experiments are presented in chapter 5

4. Make a detailed qualitative analysis of the performance on different kinds of questions. The results of this analysis are also presented in chapter 5

The rest of this chapter presents the tools and measures being used.

3.2 Building a question corpus

Since, the purpose of the present work is to re-examine previous work in the field, as well as establish what performance can be expected in the face of a random sample of real users' questions, such a corpus had to be constructed. This corpus is based on the AnswerBus logs. AnswerBus is a running on-line question answering system[1] and details about the system can be found in Zeng (2002). The authors have made 25,000 logged questions available to the public and for the purpose of the present work 5,000 were extracted randomly and manually tagged according to the taxonomy established by Li & Roth (2002) and described in detail in chapter 2.

The classes in the taxonomy essentially lack formal definitions and therefore re-using it becomes a matter of example-based classification. When confronted with a new question, one has to look up how such questions were classified in the original corpus, as marked up by Li & Roth (2002). This corpus is described in chapter 2. In short, it consists of 5,000 questions, most derived from TREC data.

This process becomes iterative in nature in that the person performing the tagging has to switch back and forth between the corpus to be tagged and the corpus tagged by the taxonomy's constructors. The understanding is to some extent gradually enhanced and modified in the process. Therefore, when the AnswerBus corpus had been tagged completely, the tagging was revised based on the final understanding of the taxonomy. Most classes are quite clear. However, some difficulties arise in the classification process, and these difficulties are presented in the next chapter, along with an informal evaluation of said taxonomy.

[1] `http://answerbus.coli.uni-sb.de/index.shtml`

3.3 Machine learning

As stated above, five different algorithms have been used throughout the rest of the work, namely Naïve Bayes, k Nearest Neighbors, Decision Tree Learning, Sparse Network of Winnows, and Support Vector Machines. In order to evaluate how these different machine learners perform in the question classification task the WEKA system for machine learning was used[2] (Witten & Frank 2000). WEKA contains implementations of most common machine learners in JAVA. The benefit of using this framework for the experiments is that the data is pre- and post-processed in the exact same way, leaving minimal variance in results between the learners based on other things than the learners themselves. However, we also opted to test the SNoW algorithm since it has been used successfully in previous work. The SNoW algorithm is not implemented within WEKA and so the original implementation of this algorithm was used[3]. No parameter tweaking has been performed, i.e. all algorithms are run with default parameter values.

3.3.1 Question representation

When dealing with machine learning techniques, it has been established that the choice of representation for a problem has great impact on the outcome of the technique at hand. As a consequence, if the representation is flawed, it does not matter which algorithm is used (cf. Rendell & Cho 1990, Mitchell 1997). The problem at hand, question classification, has many similarities with the problem of text categorization, which has been extensively studied (cf. Sebastiani 2002). Text categorization is the problem of assigning a text to one or more predefined categories, based on the content of the text. Question classification can be seen as an instance of this more general problem.

In text categorization, content has often been viewed as the following:

Definition 2 *The content of a text is (a subset of) the words that comprise the text.*

This is also known as a bag-of-words approach to text representation. In this scheme, a text is represented as a vector $\overrightarrow{t}$ of term weights, such that $\overrightarrow{t} = (w_1, w_2, \ldots, w_n)$ where w_i are the weights of the word in the text. In text classification, weights are typically calculated as the term frequency

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] http://l2r.cs.uiuc.edu/~ danr/snow.html

(i.e., the frequency of a given word token) multiplied with the inverse document frequency (calculated by dividing the total number of documents in the data with the number of documents in which a given term/word occurs).

The problem with this approach is that it disregards syntax and semantics entirely. The question 'Who shot JFK?' is seen as identical to 'JFK shot who?' and bare only little in common with 'Who murdered John F. Kennedy?'. Fortunately, in the case of question classification all of the previous questions expects the same answer type. For text categorization it has been found that representations more sophisticated than this do not yield significantly better results (cf. Sebastiani 2002).

We have limited ourselves to using unweighted term vectors in this work.

3.4 Evaluation

This section contains a complete description of the measures used to evaluate the quantitative performance of the different classifiers. It begins by defining true and false positives and negatives, which are then used to calculate precision and recall. Both micro- and macro-averaged precision and recall are then defined. The section is concluded by defining two statistical measures, two different s-tests, that are used to compare the performance of the different algorithms to each other.

Because of the subjective nature of question classification, question classifiers can not be evaluated analytically by proving their correctness and completeness. Rather, classifiers are evaluated experimentally by means of their effectiveness with respect to precision (π) and recall (ρ). Precision is also often referred to as accuracy. To calculate π and ρ, we need to define true positives, false positives, true negatives, and false negatives.

Definition 3 *Given a question q_i, if a classifier correctly assigns q_i to a category c_i, as judged by an expert, this is referred to as a true positive (TP_i).*

Definition 4 *Given a question q_i, if a classifier erroneously assigns q_i to a category c_i, as judged by an expert, this is referred to as a false positive (FP_i).*

Definition 5 *Given a question q_i, if a classifier correctly rejects q_i as belonging to a category c_i, as judged by an expert, this is referred to as a true negative (TN_i).*

Definition 6 *Given a question q_i, if a classifier erroneously rejects q_i as belonging to a category c_i, as judged by an expert, this is referred to as a false negative (FN_i).*

Two different methods for estimating precision (π) and recall (ρ) can be used: micro-averaging, denoted as $\hat{\pi}^{\mu}$ and $\hat{\rho}^{\mu}$ respectively, and macro-averaging, denoted as $\hat{\pi}^{M}$ and $\hat{\rho}^{M}$ respectively, (Sebastiani 2002). These two methods yield different results if the categories have very varying generality. Micro-averaged precision and recall is dominated by the large categories, whereas macro-averaged precision and recall illustrates how well a classifier performs across all categories. The measures are calculated as follows:

$$\hat{\pi}^{\mu} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FP_i)} \tag{3.1}$$

$$\hat{\rho}^{\mu} = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FN_i)} \tag{3.2}$$

$$\hat{\pi}^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\pi}_i}{|\mathcal{C}|} \tag{3.3}$$

$$\hat{\rho}^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\rho}_i}{|\mathcal{C}|} \tag{3.4}$$

where $|\mathcal{C}|$ is the total number of categories, and $\hat{\pi}_i$ and $\hat{\rho}_i$ are the precision and recall of category i. A system striving to reach high precision will invariably lower its recall score, and vice versa. This has lead to the use of a combined measure called the F-measure (F):

$$F = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho} \tag{3.5}$$

where π and ρ can be either micro- or macro-avereged values. F balances π and ρ by means of a weight β. When $\beta = 1$, π and ρ are given equal weight. When $\beta > 1$, precision is favored, and when $\beta < 1$, recall is favored.

3.4.1 Validation/Significance testing

A common shortcoming of many comparisons between machine learning algorithms with respect to a certain problem is that few researchers validate their results by means of statistical significance testing. In essence, this means that very few conclusions can be drawn from their work. Yang & Liu (1999) defines a number of significance tests that can be used to compare

both micro- and macro-averaged effectiveness. For present purposes only micro and macro sign tests will be used.

Micro sign test

The micro sign test (s-test) compares two systems based on the binary decisions of all pairs $\langle q_j, c_i \rangle \in \mathcal{Q} \times \mathcal{C}$, where $\mathcal{Q}$ is the domain of questions and $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ is a set of predefined categories. We define the following:

- $N = |\mathcal{Q} \times \mathcal{C}|$
- $a_{i,j} \in \{0, 1\}$ is the measure of success for system A on the pair $\langle q_j, c_i \rangle$, where 1 means correct and 0 means incorrect.
- $b_{i,j} \in \{0, 1\}$ is the measure of success for system B on the pair $\langle q_j, c_i \rangle$.
- n is the number of times $a_{i,j}$ and $b_{i,j}$ differ.
- k is the number of times $a_{i,j} > b_{i,j}$.

The null hypothesis is that $k = 0.5n$ or that k has a binominal distribution of $Bin(n, p)$ where $p = 0.5$. The alternative hypothesis is that k has a binominal distribution of $Bin(n, p)$ where $p > 0.5$, meaning that system A is better than system B.

The P-value (one-sided) can be approximated by using the standard normal distribution:

$$Z = \frac{k - 0.5n}{0.5\sqrt{n}} \tag{3.6}$$

Given a skewed category distribution, the micro sign test can be said to be dominated by the large categories.

Macro sign test (S-test)

The macro sign test (S-test) is also intended to compare two systems based on the paired F_1 values for individual categories.

- M equals the number of unique categories, $|\mathcal{C}|$.
- $a_i \in \{0, 1\}$ is the F_1 score of system A on the ith category ($i = 1, 2, \ldots, M$).
- $b_i \in \{0, 1\}$ is the F_1 score of system B on the ith category ($i = 1, 2, \ldots, M$).

- n is the number of times a_i and b_i differ.
- k is the number of times that a_i is larger than b_i.

The test hypotheses and the P-value (one-sided) computation are the same as in the micro s-test.

3.5 Summary

This chapter presented the research issues in detail, as well as gave a description how these issues are to be approached. We began by looking at how a new corpus of questions was constructed. Then we looked at what machine learning algorithms have been used, and in which implementations. Details on how questions have been represented for the machine learners were also presented. The chapter concluded by describing how the results will be evaluated, what measures are used, and how we define these measures. In the next chapter we present an evaluation of the taxonomy used throughout this work. This evaluation is necessary in order to get a complete picture of the premises on which the rest of the work rely.

4

Evaluation of taxonomy

This chapter presents an evaluation of the taxonomy created by Li & Roth (2002). As previously mentioned, this taxonomy has been chosen for the experiments conducted in the present thesis, since it has been used in several previous experiments on machine learning techniques for question classification (cf. Zhang & Lee 2003b, Hacioglu & Ward 2003).

In order to assess the taxonomy both the tagged TREC corpora made available by Li & Roth (2002) as well as a corpora based on the AnswerBus logs were used. The former was presented in detail in chapter 2 and the latter in chapter 3.

The present chapter also contains a brief analysis of the original tagging of the TREC corpus, and attempts to list decisions that affected the tagging of the AnswerBus corpus.

The purpose of the evaluation is to make clear how the taxonomy is constructed in detail and how it compares to other taxonomies in order to understand how this might affect the results derived from the experiments. The analysis of the original tagging is presented in order to understand what problems and issues arise when tagging a new corpus according to the taxonomy, and what design choices that have been made.

4.1 Taxonomies

The taxonomies used in question answering can be roughly divided into three different sizes: small taxonomies that are flat and usually consists of 6-12 different categories, medium sized taxonomies that can be both flat and hierarchical and consist of 15-30 different categories, and large taxonomies that are hierarchical and consist of 50-150 categories. Examples of flat taxonomies are found in A and hierarchical are found in B.

4.2 Category distribution

Table 4.1 shows the distribution of the different categories in the TREC and AnswerBus corpora respectively.

Class		TREC	AnswerBus
Abbreviation		*1.6%*	*1.1%*
	abb	0.3%	0%
	exp	1.3%	1.1%
Entity		*24.3%*	*9.6%*
	animal	2.1%	0.9%
	body	2.1%	0.9%
	color	0.3%	0.06%
	creative	3.8%	2.4%
	currency	0.07%	0%
	dis.med.	1.9%	0.6%
	event	1.0%	0.2%
	food	1.9%	0.6%
	instrument	0.2%	0.1%
	lang.	0.3%	0.1%
	letter	0.1%	0.06%
	other	4.0%	0.2%
	plant	0.2%	0.2%

Continued on next page

Class		TREC	AnswerBus
	product	0.8%	0.8%
	religion	0.07%	0.06%
	sport	1.1%	0%
	substance	0.8%	0.6%
	symbol	0.2%	0.2%
	technique	0.7%	0.3%
	term	1.7%	0.3%
	vehicle	0.5%	0.5%
	word	0.5%	0.5%
Description		*21.2%*	*27.0%*
	definition	7.7%	13.8%
	description	5.0%	6.0%
	manner	5.0%	7.2%
	reason	3.5%	3.0%
Human		*22.5%*	*21.9%*
	group	3.5%	1.7%
	ind	17.7%	13%
	title	0.5%	0.2%
	description	0.8%	7%
Location		*17.7%*	*20%*
	city	2.4%	1.6%
	country	2.8%	0.7%
	mountain	2.8%	0.3%
	other	8.5%	16.9%
	state	1.2%	0.5%
Numeric		*16.6%*	*17.3%*
	code	0.2%	0.6%
	count	6.7%	5.6%
	date	4.0%	5.5%
	distance	0.6%	1.8%

Continued on next page

Class	TREC	AnswerBus
money	1.3%	0.7%
order	0.1%	0.06%
other	1.0%	0.06%
period	1.4%	1.7%
percent	0.5%	0.3%
speed	0.2%	0.3%
temp	0.2%	0.3%
size	0.2%	0.06%
weight	0.2%	0.3%

Table 4.1: Hierarchical answer type taxonomies

From table 4.1 we see that the category distribution in the two corpora is rather similar. For the vast majority of categories, the distribution is about the same.

We will now go through each coarse category class to assess the taxonomy. We will do this in general terms, and also look specifically at coverage and granularity. Coverage concerns whether the super-category contains subcategories that covers all conceivable examples, while granularity concerns if the category is to broad, i.e. covers to many examples, or to narrow, i.e. covers very few and verify specific examples.

4.2.1 Abbreviation

Questions relating to abbreviations can take two forms. Either, a user is interested in how to abbreviate something (abbr:abb), or the user wants to know the correct expansion of an abbreviation (abbr:exp).

Coverage The abbreviation category covers the two types of questions that occur in relation to abbreviations in both the TREC and the AnswerBus corpora. It is perhaps interesting to note that there are no questions on how to abbreviate something in the AnswerBus corpus.

Granularity There is no need to break down either of the two subcategories into finer ones.

4.2.2 Entity

The difference in relative distribution between the corpora with regards to the entity class is notable. There are consistently an equal number of, or more, occurrences of the subcategories with regards to the overall corpus within the TREC corpus than within the AnswerBus corpus. This might reflect the fact that the TREC corpus is slightly biased due to manual selection of questions.

Coverage In the entity category we find a category for musical instruments and vehicles, but also a category for products in general. It is not clear why the former have generated specialized categories, when one could easily imagine others, like furniture and weapons, that do not have dedicated categories. We also find the categories word and term which are not always easily distinguishable.

Granularity The entity class would perhaps benefit from having subcategories. For instance, in the taxonomy proposed by Suzuki, Taira, Sasaki & Maeda (2003) (see appendix B) a category such as vehicle is a subcategory of the category product, which in turn is subdivided into e.g. car and train.

4.2.3 Description

For the description class questions belonging to the desc:def are twice as frequent within the AnswerBus corpus than within the TREC corpus. It is hard to assess the description category in terms of coverage and granularity, since it isn't exactly clear what it intends to cover. It seems to be the category that deals with answers that are not just a simple noun phrase, but rather some elaboration.

Coverage Description seems to cover most questions that require detailed elaborations as answer quite well.

Granularity It is not always clear how to distinguish between the subclasses description and manner, and the latter also partially overlaps the category for techniques and methods in the entity category.

4.2.4 Human

In general, certain questions are difficult to categorize under the given taxonomy, such as ”Who invented the post-it note?”. It is not entirely clear if we expect a specific person or an organization as the answer. Interesting to note is that the number of questions belonging to the subcategory hum:desc in the AnswerBus corpus is far greater than in the TREC corpus. 7% as opposed to 0,8%.

Coverage The human category covers the kinds of questions asked in relation to humans and organizations.

Granularity The human category could benefit from having subcategories. Almost one quarter of all questions in the AnswerBus corpus concerns this category. The taxonomies proposed by Hovy et al. (2002) and Suzuki, Taira, Sasaki & Maeda (2003) (see appendix B) both have finer granularity with regards to persons and organizations. In the group category, one could for instance differentiate between companies, military organizations, and political parties.

4.2.5 Location

The most interesting thing to notice in the location class is the difference in numbers with regards to the loc:other category. We will look further into this specific issue in the next section.

Coverage The location category covers all questions concerning locations by introducing a subcategory other.

Granularity The subcategories in the location class seems rather ad hoc. Why is there a category for mountains but not lakes or rivers? There are several question related to the latter two in the TREC corpora so the reason seems unclear. The fact that the single largest category in the AnswerBus corpus is the loc:other category also indicates that this coarse class could benefit from a larger set of subcategories.

As with the human category, the taxonomies proposed by Hovy et al. (2002) and Suzuki, Taira, Sasaki & Maeda (2003) both have finer granularity with regards to locations.

4.2.6 Numeric

The numeric category covers questions concerning all kinds of numeric information, such as dates, ages, and speed. The distribution within this class is roughly the same in both the TREC and AnswserBus corpora.

Coverage In the number, as in the location category we find a subcategory other which ensures that the coverage is perfect. Once again this is perhaps not a satisfactory solution, and Hovy et al. (2002) and Suzuki, Taira, Sasaki & Maeda (2003) have no such "fallback" category.

Granularity In the number category, the subclasses code could for instance be split up into zip codes and social security numbers and size into area and volume.

4.3 Remarks on original tagging and category usage

This section presents a review of the original tagging in the TREC corpus. We look at cases were the choice of tag is not entirely clear cut and also what design choices were made in the tagging of the AnswerBus corpus.

4.3.1 Abbreviation

In the TREC material, "What is BPH?" is tagged as abb:exp and "What is Plc?" is tagged as desc:def. It is unclear how one can differentiate between these two questions, and in the AnswerBus corpus the former is classified as desc:def. The logic being that a definition of the term is more probable to be satisfactory for the user, and is also likely to contain an expansion of the abbreviation.

4.3.2 Entity

The question "What's the Hungarian word for pepper?" and "What is another word for diet?" are tagged as enty:word while "How do you say 'fresh' in Spanish?" and "What are dinosaur droppings called?" are tagged enty:termeq. enty:word are described as "words with a special property" and enty:termeq as "equivalent terms". Here the problem with two categories not being easily distinguishable becomes apparent. In the AnswerBus corpus all these questions would be treated as enty:termeq.

The question "What is the recipe for or formula for Coca-Cola?" is tagged as enty:food while "What are the ingredients of Coca-Cola?" is tagged as enty:substance. In the AnswerBus corpus both these questions would be treated as enty:food.

"What do you call a group of geese?" is tagged as enty:animal, while the question "What is the collective noun for geese'?" is tagged as enty:word. In the AnswerBus corpus such questions are tagged as enty:word.

Tagged as enty:product and not as enty:vehicle we find "What car was driven in the 199 release of 'Smokey and the Bandit'?". In AnswerBus, this would have been classified as enty:vehicle.

While "What is the quickest and easiest way to get nail polish out of clothes?" is tagged as enty:techmeth, "How can you get rust stains out of clothing?" is tagged under a whole different coarse category as desc:manner. Admittedly there are differences between these two categories, and the same distinction is made in the AnswerBus corpus.

4.3.3 Location

When reviewing the tagging performed on the TREC material and when categorizing new material the most apparent problem comes in categorizing questions like "Where can i find information about X?". In the original tagging, these are tagged as other in the location class. They therefore warrant the same treatments as rivers, lakes and planets. One could argue that they are a special form of location and could warrant their own category. In fact these questions are more like traditional search queries with the phrase "where can i find information about" prefixed to them - they expect a list of URL:s rather than a specific answer. Furthermore, in the TREC material they constitute about 8% of the questions and in the AnswerBus corpus 16%, which also indicates that they might benefit from being treated as a special form of question.

4.3.4 Description, Human, and Numeric

Concerning the categories description, human, and numeric nothing remarkable was found. These categories seem to be straightforward to apply to a new corpus.

4.4 Yes/no-questions

Perhaps the most remarkable aspect about the taxonomy is the lack of a class for simple yes/no-questions. A question like "Is the stomach a muscle?" therefore poses certain problems. However, the lack of this category might be due to the fact that the TREC material used as a basis for the taxonomy contains no such questions. The AnswerBus corpus on the other hand contains plentiful yes/no-questions. However, the omission of the category can also be a conscious decision, perhaps the system designers do not want the system to answer simply "yes" or "no", but rather to provide the user with a more detailed and elaborate answer.

4.5 Summary

In this chapter we have scrutinized the taxonomy used throughout the present work. We saw that the taxonomy seem to be bootstrapped to the TREC corpus for which it was developed. The most clear evidence for this is perhaps the lack of a yes/no-category, and we also saw that in many cases the choice of category is not entirely clear. The next chapter presents the actual results obtained from running the different machine learning algorithms on the different corpora.

5 Experiments

This chapter presents the results from the experiments on question classification that are main piece of work in the thesis. First, we look at a re-examination of the previous work in the field. Next we look at how the algorithms perform on the AnswerBus corpus. We conclude by looking closer at what questions that pose the biggest problems for the algorithms.

5.1 Study 1: Re-examining previous work on question classification

The first experiment is a straightforward re-examination of how the five machine learning algorithms that are the focus of the present work perform and how they relate to each other in terms of performance. The algorithms are, as mentioned in chapter 3, *k* nearest neighbors (kNN), naive bayes (NB), decision tree learning (DT), sparse network of winnows (SNoW), and support vector machines (SVM).

This experiment has been done under two different settings. First, we have used the corpus originally developed by (Li & Roth 2002), but since that test corpus consists of questions solely from TREC-10 and the TREC conferences have a specific agenda, the test corpus might be slightly different from the training data. Therefore, a second setting was used where the questions from the training and test corpora were pooled together and a

randomized test corpus was extracted. This will be referred to as the repartitioned corpus. The five learning algorithms were run on both corpora.

The performance of the different learners in the first setting can be found in table 5.1, results from the second setting are found in table 5.2 while significance testing between the learners is shown in table 5.3. As mentioned in chapter 3 all algorithms were run with default parameters, and questions were represented as bag-of-words. In the tables below, π^{μ}, ρ^{μ}, F_1^{μ} refers to micro-averaged precision, recall, and F_1 score respectively, while π^{M}, ρ^{M}, and F_1^{M} refers to the corresponding macro-averaged scores.

Classifier	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
kNN	.67	.67	.67	.60	.50	.55
NB	.72	.71	.71	.60	.58	.59
SNoW	.76	.75	.76	.71	.64	.67
DT	.78	.78	.78	.75	**.68**	**.71**
SVM	**.81**	**.81**	**.81**	**.76**	.67	**.71**

Table 5.1: Performance of classifiers on original TREC data.

In the tables, the highest score of each column is highlighted in boldface. Looking at results of the algorithms on the original TREC corpus in table 5.1 we see that the SVM algorithm has the highest performance scores in all calculations but the macro-averaged recall. SVM and Decision Trees ties in the macro-averaged F_1 score.

Classifier	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
kNN	.63	.63	.62	.62	.56	.59
NB	.67	.67	.67	.60	.65	.62
SNoW	.66	.66	.66	.65	.50	.57
DT	.72	.72	.72	.64	.62	.63
SVM	**.78**	**.78**	**.78**	**.81**	**.71**	**.76**

Table 5.2: Performance of classifiers on repartitioned TREC data.

Looking at the results from the experiments on the repartitioned corpus in table 5.2 we see that the SVM outperforms all the other algorithms in all calculations.

As can be seen in table 5.1 and 5.2 the performance of the different learning algorithms with regards to micro-averaged precision and recall is at best equal to and in most cases worse on the repartitioned data than on the original data. This suggests that the test data in the first setting does not reflect the training data as well as one would want, i.e. the results are overly positive in relation to what could be expected. This indicates that all previous work based on this corpus reports over-optimistic results. When it comes to macro-averaged precision and recall the results are more varied and it is difficult to spot a clear pattern.

		Original		Repartitioned	
sysA	sysB	s-test	S-test	s-test	S-test
kNN	NB	$<$	-	$<$	$\ll$
kNN	SNoW	$\ll$	$<$	-	-
kNN	DT	$\ll$	$\ll$	$\ll$	$\ll$
kNN	SVM	$\ll$	$\ll$	$\ll$	$\ll$
NB	SNoW	$<$	-	-	-
NB	DT	$\ll$	$\ll$	$\ll$	-
NB	SVM	$\ll$	$\ll$	$\ll$	$\ll$
SNoW	DT	$<$	-	$\ll$	-
SNoW	SVM	$\ll$	-	$\ll$	$\ll$
DT	SVM	$\ll$	-	$\ll$	$\ll$

Table 5.3: Significance testing of classifiers on both original and repartitioned TREC data.

In table 5.3 we can find differences when comparing the algorithms with regards to significant differences in performance. In the table, "$<$" means a difference on the .05 significance level, and "$\ll$" means a difference on the .01 level. So, NB $<$ SNoW should be read as NB performs significantly worse than SNoW on the .05 level. The column "s-test" means micro sign test, and "S-test" means macro sign test (see section 3.4.1 in chapter 3 for details on the tests).

It is interesting to note that where there were no significant differences in performance on the original corpus there are to some extent differences on the repartitioned corpus and also the other way around to a smaller extent. For instance, the SVM is significantly better than both decision trees and SNoW with regards to the S-test in the repartitioned data as opposed

to the original data. This suggests that the training and test corpora in fact are not balanced in the original setting, and some of the results reported in previous work are somewhat biased. If we look at the results from the repartitioned corpus we could order the algorithms from better to worse according to the following for the s-test scores: SVM $\gg$ DT $\gg$ {NB, SNoW, kNN}.

If we then turn to the S-test scores the results are not as clear, but we can propose the following ordering: SVM $\gg$ {DT, NB} $>$ SNoW $>$ kNN. SNoW is difficult to place in this ordering.

5.2 Study 2: The AnswerBus corpus

To further investigate the performance of different machine learners in the face of a corpus consisting of actual users' questions a second experiment was conducted. In this setting 5,000 questions from the AnswerBus logs were used (see chapter 3 for details), but everything else remains the same as in experiment 1. The corpus is split up into a training corpus consisting of 4,500 questions, and a test corpus of 500 questions. Results in terms of performance are found in table 5.4 and significance testing between the classifiers are found in table 5.5.

Classifier	π^{μ}	ρ^{μ}	F_1^{μ}	π^{M}	ρ^{M}	F_1^{M}
kNN	.72	.71	.71	.61	.53	.56
NB	.80	.80	.80	.70	.66	.68
SNoW	.69	.66	.67	.61	**.77**	.68
DT	.81	.80	.80	.69	.66	.67
SVM	**.82**	**.81**	**.82**	**.73**	.65	**.69**

Table 5.4: Performance of classifiers on AnswerBus data.

As can be seen in table 5.4 the performance in terms of micro-averaged precision and recall is higher on the AnswerBus corpus than on any of the TREC corpora. When it comes to macro-averaged performance the results are more varied and it is hard to draw any clear conclusions. We can see that the SVM outperforms all other algorithms in all cases but for macro-averaged recall, where the SNoW algorithm excels.

In terms of significant differences between classifiers, the results from the AnswerBus corpus deviates from what could have been expected given

		AnswerBus	
sysA	sysB	s-test	S-test
kNN	NB	$\ll$	$<$
kNN	SNoW	-	-
kNN	DT	$\ll$	$\ll$
kNN	SVM	$\ll$	$\ll$
NB	DT	-	-
NB	SVM	-	-
SNoW	NB	$\ll$	-
SNoW	DT	$\ll$	-
SNoW	SVM	$\ll$	-
DT	SVM	-	-

Table 5.5: Significance testing of classifiers on AnswerBus data.

the results on the TREC corpora (both original and repartitioned). It seems that Naïve Bayes, Decision Trees and Support Vector Machines are on par with each other, while *k* Nearest Neighbors and Sparse Network of Winnows are significantly worse in terms of performance when it comes to micro-averaged scores. For macro-averaged scores the results are not as clear. We see that the kNN is significantly worse than SVM, DT and NB, but no other differences between the algorithms are found. It might be the fact that the category distribution is extremely skewed, i.e. there are very many occurences of questions belonging to a small number of the categories, while there are many categories with very few examples in the corpus. The small categories, which are in majority, tend to dominate the macro-averaged scores.

5.3 Analysis of Problematic Questions

This section presents a more detailed study of those questions that the machine learners had problems classifying. We will here restrict our study to the AnswerBus corpus. Out of 500 test questions, there were 14% which all the machine learning algorithms failed to classify. These questions will be the focus of the analysis below.

After looking at the questions, the following possible sources of error was established:

- Problems due to small categories, **21%**
- Problems due to the processing and representation of questions, **25%**
- Problems due to the taxonomy per se and the usage of categories, **4%**
- Problems due to over-generalization, **9%**
- Problems due to spelling errors, **3%**
- Problems of unknown origin, **38%**

5.3.1 Small categories

Looking at those questions that were not classified correctly by any of the machine learning algorithms, we see that 21% belong to categories that constitute less than 1% of the data in the AnswerBus corpus, namely enty:product, enty:food, enty:dismed, and enty:other.

Given the skewed category distribution in the corpus, this is an expected problem. Some categories have too few examples entirely for the algorithms to be able to generalize over. Building a larger corpus should to a large extent minimize this problem. Another approach would be to go through these categories and see if they can be merged into larger ones, or be incorporated in other categories.

5.3.2 Variations in formulation

The questions were represented as bags-of-words in the form of feature vectors. This means that plural and singular forms och nouns are unrelated in the view of the learning algorithm. Also, synonyms are seen as unrelated, since no semantics is incorporated. 25% of the errors seem to stem from this problem. For instance, "What countries are in the European Union?" contain the plural of "country", and only the singular version occurs in the training corpus. This is related to the problem with small categories, and essentially becomes a problem of sparse data. Unlike the problem with small categories, however, this problem might be overcome by using stemming algorithms, and other pre-processing such as named entity recognition. The research conducted by Li & Roth (2002) suggests that this is a viable approach.

5.3.3 Taxonomy and usage

One source of confusion for the learning algorithms is questions that are superficially very similar, but are tagged as different categories in the corpus. 4% of the problematic questions seem to be misclassified due to this fact. One typical example is the question "What was Roy Roger's dog's name?" which, if we use Li and Roth's (2002) tagging of the TREC corpus as a standard, belongs to the enty:animal category. This question is very similar to a question such as "What was Mao's second name?", at least from the perspective of a machine learning algorithm using a bag-of-words representation of questions. However, the latter question belongs to the hum:ind category.

5.3.4 Keyword errors

Some questions (9%) that were classified erroneously contain specific words that the learners are susceptible to generalize to a single specific category. The two questions "What's the longest a dog has ever lived?" and "What was the longest human pregnancy?" both contain the word "longest", which often occurs in questions related to the num:dist category.

All machine learning algorithms attempt to generalize over the data as much as possible. In the context of question classification and text categorization, this means that specific words, or sequences of words becomes associated to specific categories to some extent. We can expect this to be a problem even if we up-size the training corpus. Some questions are too similar on the surface for a learning algorithm to be able to find a pattern for discerning between them.

5.3.5 Spelling errors

Spelling errors are a serious problem for machine learning algorithms. A total of 3% of the questions that failed to be categorized contains a spelling error that, if corrected should lead to a correct classification.

In essence, machine learning algorithms has low tolerance when it comes to the input being corrupt in any form. Any word that has not been seen by the algorithm before is disregarded as input. If the question stem or question word itself is corrupt the question becomes almost impossible to categorize, since this is an essential piece of information for most categories of questions.

5.3.6 Errors of unclear origin

A large portion of the question that are classified incorrectly by all the learners seem to be more difficult to explain. On the surface they are similar to other questions in the same category that have been categorized correctly. Almost half of these, however, belong to the desc:desc category. Most of them are incorrectly categorized as either desc:def or desc:manner. A total of 38% of the questions that were misclassified by all of the algorithms are difficult to explain.

5.4 Summary

In this chapter the results from the experiments were reported. We saw that by repartitioning the TREC data we received different results than when running the TREC data in its original setting, both in terms of raw and relative performance. We then saw that using the AnswerBus data yielded even bigger differences in terms of the relative performance between the classifiers. We saw that naive bayes, decision trees and support vector machines performed on par with each other in regards to micro-averaged performance, while all algorithms but the k nearest neighbors were equal in performance on the macro-averaged results. We also looked at questions that are problematic for machine learners, and tried to establish possible reasons for this. In the next chapter we will discuss the results presented here.

6 Conclusions and future work

This chapter contains a discussion of the results obtained from the experiments presented in the previous chapter and points to future work.

6.1 Study 1

By repartitioning the TREC-based training and test corpus, we could see that not only did the absolute performance of the individual algorithms change, but also the relative performance between them. More specifically, apparent significant differences between different algorithms disappeared and new ones emerged, especially in relation to the macro sign test. This indicates that the TREC corpus indeed is biased. And a reasonable explanation for this is that the test corpus stems from the TREC-10, while the training corpus stems from other sources, TREC and other.

An alternative explanation is that the corpus is simply too small. Since the category distribution is skewed (i.e. some categories are very frequent, while others are quite infrequent), we might end up with very few examples of a category in the training corpus, leading to difficulties for the classifiers to learn these. The best solution here would be to run a ten-fold cross-validation in order to establish more accurate results. Performing ten-fold cross validation is a standard technique to achieve more accurate results (Sebastiani 2002). In a ten-fold cross-validation we would pool the existing

training and test material and divide it into ten parts of equal size. Ten training and classification iterations would then be run, the first using the first tenth as test corpus and the rest as training corpus, the second using the second tenth as test corpus and the rest as training corpus, and so on over the complete set. We would then average (or calculate some other kind of combined measure) over the results in order to establish more accurate results.

The results from the experiment on the repartitioned corpus suggest that in regards to the micro sign test the SVM outperforms all other learning algorithms, and DT outperforms all algorithms but the SVM algorithm. The kNN, SNoW and NB seem to perform on par with each other.

Looking at the results from the macro sign test, SVM is still the best performer, while DT and NB seem to perform on par with each other, and SNoW and kNN are roughly on the same level of performance.

The algorithm which performance was most severely affected in terms of absolute performance by repartitioning the corpus was the SNoW classifier. It was also this algorithm that caused most differences in the significance comparisons. The exact cause for this is unclear. It is unfortunate that this is the only algorithm that is not implemented in the WEKA system, and hence a standalone implementation has been used. This means that SNoW is more susceptible for variation due to external factors, such as pre-processing.

6.2 Study 2

In the second experiment a corpus based solely on real users' questions was used. The intention was to establish what could be expected in terms of performance of machine learning techniques in a real on-line system. We can see that the performance in terms of micro-averaged results are consistently higher than what could be expected from the TREC corpus. Given that the category distribution in the AnswerBus corpus is even more skewed than in the TREC corpus, i.e. there are very many occurrences of a few categories, this might be expected. Most notable are the HUM:desc as exemplified by questions such as "Who is XXX?" and LOC:other and the very common question "Where can I find information about XXX?". These two categories alone represents almost 25% of the corpus and are quite easy to categorize for all algorithms.

When we turn to the macro-averaged results, it is hard to draw any clear conclusions. Overall the performance on the AnswerBus corpus is on

par with the performance on the repartitioned TREC corpus, but for Naïve Bayes precision goes up 10% and for SNoW recall increases by 27%. The conclusion that could be drawn is that some algorithms are more sensitive to skewed category distributions.

Turning to the relative performance of the algorithms on the AnswerBus corpus as compared to the repartitioned TREC corpus we find a few interesting results. The ranking in terms of the results from the micro sign test differs strongly from what might be expected from the TREC material. Now, three algorithms perform on par with each other, namely SVM, DT and NB. Given that these are equal in performance other factors becomes more interesting in choosing an algorithm to use in an on-line system. Algorithms that are fast in the classification phase are to be preferred to slower ones, given all else equal.

In terms of the results from the macro sign test, the performance of the algorithms are roughly in line with what could be expected from the results on the repartitioned TREC corpus.

6.3 The overall picture

Looking at the overall picture, taking all the results into account, it is hard to draw any absolute conclusions. On the one hand, the SVM algorithm is significantly better than all the other algorithms as far as performance on the repartitioned TREC corpus goes. However, on the AnswerBus corpus, DT and NB perform just as good as the SVM algorithm. Taking into account that the AnswerBus corpus is more skewed in terms of the category distribution than the TREC counterpart, there are fewer examples of the small categories in absolute terms in the AnswerBus corpus. Hence, there might be too few examples for a machine learning algorithm to even be theoretically able to learn to discern between them. Real differences between algorithms might not show up because of this, and given a larger corpus, the same ranking between algorithms that exist on the TREC material might emerge.

6.4 Future work

The most obvious way to proceed is to build a larger question corpus. In this work we have used 5,000 of the 25,000 questions available in the AnswerBus logs. It would be a straightforward task to classify more of this material. This seems to be a necessary task since the category distribution

is extremely skewed. There are many categories that have very few occurrences. Perhaps too few for any lerning algorithm to be able to learn to descern between them. It would also be interesting to run a ten-fold cross validation in order to get more accurate and reliable results.

This work has used one taxonomy exclusively, and it would be interesting to run the learning algorithms on a corpus tagged according to another, perhaps more general, taxonomy.

A

Flat answer type taxonomies

Table A.1: Flat answer type taxonomies

Author(s)	**Total types**	**Type taxonomy**
Eichmann & Srinivasan (1999)	4	DATE MONEY NUMBER NAME
Litkowski (1999)	6	TIME LOCATION WHO WHAT NUMBER SIZE
Oard et al. (1999)	6	PERSON TIME/DATE LOCATION NUMBER AMOUNT ORGANIZATION
Pinto et al. (2002)	7	PERSON ORGANIZATION

Continued on next page

Author(s)	Total types	Type taxonomy
		PERCENT TIME LOCATION DATE MONEY
Singhal et al. (1999)	8	PERSON LOCATION DATE QUANTITY ORGANIZATION DURATION LINEAR_MEASURE OTHER
Ittycheriah et al. (2000)	9	PERSON ORGANIZATION DATE PERCENTAGE LOCATION TIME MONETARY VALUE PHRASE REASON
Hull (1999)	10	PERSON PLACE TIME MONEY NUMBER QUANTITY NAME HOW WHAT UNKNOWN
Wu et al. (2000)	11	PERSON LOCATION ORG MONEY PERCENTAGE DATE TME DURATION LENGTH SIZE NUMBER

Continued on next page

Author(s)	Total types	Type taxonomy
Laszlo et al. (1999)	13	PNOUN TIMEPOINT TIMESPAN MEANS REASON MONEY CARDINAL LINEAR PERCENTAGE AREA VOLUME MASS UNKNOWN
Moldovan et al. (1999)	15	MONEY NUMBER DEFINITION TITLE NNP UNDEFINED PERSON ORGANIZATION DATE LOCATION MANNER TIME DISTANCE PRICE REASON
Radev et al. (2002)	17	PERSON PLACE DATE NUMBER DEFINITION ORGANIZATION DESCRIPTION ABBREVIATION KNOWNFOR RATE LENGTH MOINEY REASON DURATION PURPOSE

Continued on next page

Author(s)	Total types	Type taxonomy
		NOMINAL
		OTHER
Prager et al. (1999)	20	PERSON
		PLACE
		MONEY
		LENGTH
		ROLE
		ORGANIZATION
		DURATION
		AGE
		TIME
		DATE
		YEAR
		VOLUME
		AREA
		WEIGHT
		NUMBER
		METHOD
		RATE
		NAME
		COUNTRY
		STATE
Ogden et al. (1999)	27	LINEAR-SIZE
		AREA
		VOLUME
		LIQUID-VOLUME
		MASS
		RATE
		PRESSURE
		ELECTRICITY
		ENERGY
		VELOCITY
		ACCELERATION
		TEMPERATURE
		COMPUTER-MEMORY
		POPULATION-DENSITY
		TEMPORAL-OBJECT
		TIME-OBJECT
		AGE
		NAME-HUMAN
		ORGANIZATION
		PLACE
		NATIONALITY

Continued on next page

Author(s)	Total types	Type taxonomy
		INHABITANT MATERIAL EVENT-NAME PRODUCT-TYPE NUMERIC-TYPE DATE

B

Hierarchical answer type taxonomies

Table B.1: Hierarchical answer type taxonomies

Author(s)	Total types	Type taxonomy
Breck et al. (1999)	17	ANSWER +NUMERIC +-TIME +–DATE +-MEASURE +–DISTANCE +–DURATION +–RATE +ENTITY +-LOCATION +–COUNTRY +–CITY +-PERSON +–MALE +–FEMALE +-ORGANIZATION +–COMPANY
Ferret et al. (1999)	17	PERSON
		Continued on next page

Author(s)	Total types	Type taxonomy
		ORGANIZATION LOCATION -CITY -PLACE TIME-EXPRESSION -DATE -TIME -AGE -PERIOD NUMBER -LENGTH -VOLUME -DISTANCE -WEIGHT -PHYSICS -FINANCIAL
Takaki (2000)	28	PROPER -PERSON –CHAIRMAN –LEADER –MINISTER –PRESIDENT –SECRATARY –SPECIALIST -LOCATION –CITY –COUNTRY –STATE -COMPANY -LAKE -RIVER -MOUNTAIN -LANGUAGE NUMBER -SIZE -LENGTH -MONEY -PERCENT -PERIOD TIME -DATE -YEAR UNDEFINED-PROPER

Continued on next page

Author(s)	Total types	Type taxonomy
		UNDEFINED
Li & Roth (2002)	56	ABBREVIATION -ABB -EXP ENTITY -ANIMAL -BODY -COLOR -CREATIVE -CURRENCY -DIS.MED. -EVENT -FOOD -INSTRUMENT -LANG -LETTER -OTHER -PLANT -PRODUCT -RELIGION -SPORT -SUBSTANCE -SYMBOL -TECHNIQUE -TERM -VEHICLE -WORD DESCRIPTION -DEFINITION -DESCRIPTION -MANNER -REASON HUMAN -GROUP -IND -TITLE -DESCRIPTION LOCATION -CITY -COUNTRY -MOUNTAIN -OTHER -STATE

Continued on next page

Author(s)	Total types	Type taxonomy
		NUMERIC -CODE -COUNT -DATE -DISTANCE -MONEY -ORDER -OTHER -PERIOD -PERCENT -SPEED -TEMP -SIZE -WEIGHT
Hovy et al. (2002)	196 (148)	R-LOCATION -R-LOCATION -R-CAPITAL-PLACE R-TIME -R-EVENTS -R-TIME-FIRST -R-BIRTHDAY R-POPULATION -R-POPULATION R-PERSONS -R-INVENTORS R-DISCOVERERS R-POSITIONS R-WHY-FAMOUS -R-PROFESSION-TITLE -R-FIRST-PERSON R-ABBREVIATION -R-ABBREVIATION-EXPANSION -R-ABBREVIATION R-DEFINITIONS -R-DEFINITIONS A-WHY-FAMOUS -A-WHY-FAMOUS-PERSON A-DEFINITION A-TERMINOLOGY -A-TERMINOLOGY-COLLECTIVE-TERM A-ABBREVIATION-EXPANSION A-ABBREVIATION A-SYNONYM

Continued on next page

Author(s)	Total types	Type taxonomy
		A-CONTRAST A-POPULATION A-VERACITY -A-YES-NO-QUESTION -A-TRUE-FALSE-QUESTION A-TRANSLATION A-FUNCTION A-COMPONENTS A-CAUSE-OF-DEATH A-PHILOSOPHICAL-QUESTION C-TEMP-LOC -C-DATE –C-DATE-WITH-YEAR —C-DATE-WITH-YEAR-AND-DAY-OF-THE-WEEK –C-DATE-WITH-DAY-OF-THE-WEEK —C-DATE-WITH-YEAR-AND-DAY-OF-THE-WEEK -C-YEAR-RANGE –C-DECADE –C-CENTURY –C-MILLENNIUM -C-TEMP-LOC-WITH-YEAR –C-DATE-WITH-YEAR -C-DATE-RANGE -C-TIME C-AT-LOCATION C-PROPER-NAMED-ENTITY -C-PROPER-PERSON -C-PROPER-DYNASTY -C-PROPER-LANGUAGE -C-PROPER-ANIMAL -C-PROPER-PLACE –C-CONTINENT –C-WORLD-REGION –C-US-REGION –C-PROPER-COUNTRY –C-PROPER-STATE-DISTRICT –C-PROPER-COUNTY –C-PROPER-CITY –C-PROPER-CITY-DIVISION –C-PROPER-BODY-OF-WATER —C-PROPER-OCEAN —C-PROPER-SEA —C-PROPER-LAKE
		Continued on next page

Author(s)	Total types	Type taxonomy
		—C-PROPER-RIVER
		—C-PROPER-CREEK
		—C-PROPER-CANAL
		—C-PROPER-GULF
		—C-PROPER-BAY
		—C-PROPER-STRAIT
		–C-PROPER-ISLAND
		–C-PROPER-CANYON
		–C-PROPER-VALLEY
		–C-PROPER-MOUNTAIN
		–C-PROPER-VOLCANO
		–C-PROPER-DESERT
		–C-PROPER-FOREST
		–C-PROPER-STAR-CONSTELLATION
		—C-ZODIACAL-CONSTELLATION
		–C-PROPER-STAR
		–C-PROPER-PLANET
		–C-PROPER-MOON
		–C-PROPER-AMUSEMENT-PARK
		–C-PROPER-HOTEL
		–C-PROPER-PALACE
		–C-PROPER-MUSEUM
		–C-PROPER-BANK-COMPANY
		–C-PROPER-UNIVERSITY
		–C-PROPER-COLLEGE
		–C-PROPER-AIRPORT
		-C-PROPER-ORGANIZATION
		–C-PROPER-SPORTS-TEAM
		—C-PROPER-SOCCER-SPORTS-TEAM
		—C-PROPER-BASKETBALL-SPORTS-TEAM
		—C-PROPER-BASEBALL-SPORTS-TEAM
		—C-PROPER-ICE-HOCKEY-SPORTS-TEAM
		—C-PROPER-AMERICAN-FOOTBALL-SPORTS-TEAM
		–C-PROPER-CENTRAL-BANK
		–C-PROPER-POLITICAL-PARTY
		–C-PROPER-COMPANY
		—C-PROPER-BROADCASTING-COMPANY
		—C-PROPER-NEWSPAPER
		—C-PROPER-MAGAZINE
		—C-PROPER-HOTEL
		—C-PROPER-FINANCE-COMPANY
		—-C-PROPER-BANK-COMPANY
		—C-PROPER-AUTOMOBILE-COMPANY

Continued on next page

Author(s)	Total types	Type taxonomy
		—C-PROPER-AIRLINE-COMPANY —C-PROPER-OIL-COMPANY —C-PROPER-OPERA-COMPANY —C-PROPER-BALLET-COMPANY —C-PROPER-THEATER-COMPANY –C-GOVERNMENT-AGENCY C-PLANT-FLORA -C-FLOWER -C-TREE C-SUBSTANCE -C-SOLID-SUBSTANCE –C-METAL -C-LIQUID –C-BEVERAGE -C-GAS-FORM-SUBSTANCE C-QUANTITY -C-MONETARY-QUANTITY -C-SPATIAL-QUANTITY –C-DISTANCE-QUANTITY –C-AREA-QUANTITY –C-VOLUME-QUANTITY -C-TEMPORAL-QUANTITY -C-SPEED-QUANTITY -C-ACCELERATION-QUANTITY -C-NUMERICAL-QUANTITY/I-ENUM-CARDINAL -C-FREQUENCY-QUANTITY -C-SCORE-QUANTITY -C-PERCENTAGE -C-TEMPERATURE-QUANTITY -C-INFORMATION-QUANTITY -C-MASS-QUANTITY -C-POWER-QUANTITY -C-ENERGY-QUANTITY -C-MAGNETIC-FIELD-QUANTITY -C-INDUCTANCE-QUANTITY -C-RESISTANCE-QUANTITY -C-FORCE-QUANTITY -C-CHARGE-QUANTITY -C-PRESSURE-QUANTITY -C-POTENTIAL-QUANTITY -C-ILLUMINATION-QUANTITY -C-CAPACITANCE-QUANTITY -C-CURRENT-QUANTITY
		Continued on next page

Author(s)	Total types	Type taxonomy
		-C-RADIATION-QUANTITY -C-MAGNETIC-FLUX-QUANTITY C-UNIT -C-MONETARY-UNIT C-LOCATOR -C-PHONE-NUMBER -C-ADDRESS -C-ZIP-CODE -C-EMAIL-ADDRESS -C-URL C-UNIVERSITY-AGENCY C-SPIRITUAL-BEING C-OCCUPATION-PERSON C-ANIMAL C-HUMAN-FOOD C-BODY-PART C-TEMPORAL-INTERVAL -C-DAY-OF-THE-WEEK -C-MONTH-OF-THE-YEAR -C-SEASON C-DISEASE C-INSTRUMENT C-MUSICAL-INSTRUMENT C-SPORT C-LEFT-OR-RIGHT C-COLOR C-NATIONALITY S-NP S-NOUN S-VP S-PROPER-NAME ROLE REASON ROLE MANNER SLOT TITLE-P TRUE SLOT QUOTE-P TRUE SLOT POSSIBLE-REASON-P TRUE LEX SURF
Suzuki, Taira, Sasaki & Maeda (2003)	150	!TOP -NAME –PERSON —*LASTNAME —*MALE_FIRSTNAME

Continued on next page

Author(s)	Total types	Type taxonomy
		—*FEMALE_FIRSTNAME –ORGANIZATION —COMPANY —*COMPANY_GROUP —*MILITARY —INSTITUTE —*MARKET —POLITICAL_ORGANIZATION —-GOVERNMENT —-POLITICAL_PARTY —-PUBLIC_INSTITUTION —GROUP —-!SPORTS_TEAM —*ETHNIC_GROUP —*NATIONALITY –LOCATION —GPE —-CITY —-*COUNTY —-PROVINCE —-COUNTRY —REGION —GEOLOGICAL_REGION —-*LANDFORM —-*WATER_FORM —-*SEA —*ASTRAL_BODY —-*STAR —-*PLANET —ADDRESS —-POSTAL_ADDRESS —-PHONE_NUMBER —-*EMAIL —-*URL –FACILITY —GOE —-SCHOOL —-*MUSEUM —-*AMUSEMENT_PARK —-WORSHIP_PLACE —-STATION _TOP —-*AIRPORT —-*STATION
		Continued on next page

Author(s)	Total types	Type taxonomy
		—-*PORT
		—-*CAR_STOP
		—LINE
		—-*RAILROAD
		—-!ROAD
		—-*WATERWAY
		—-*TUNNEL
		—-*BRIDGE
		—*PARK
		—*MONUMENT
		–PRODUCT
		—VEHICLE
		—-*CAR
		—-*TRAIN
		—-*AIRCRAFT
		—-*SPACESHIP
		—-!SHIP
		—DRUG
		—*WEAPON
		—*STOCK
		—*CURRENCY
		—AWARD
		—*THEORY
		—RULE
		—*SERVICE
		—*CHARCTER
		—METHOD_SYSTEM
		—ACTION_MOVEMENT
		—*PLAN
		—*ACADEMIC
		—*CATEGORY
		—SPORTS
		—OFFENCE
		—ART
		—-*PICTURE
		—-*BROADCAST_PROGRAM
		—-MOVIE
		—-*SHOW
		—-MUSIC
		—PRINTING
		—-!BOOK
		—-*NEWSPAPER
		—-*MAGAZINE

Continued on next page

Author(s)	Total types	Type taxonomy
		–DISEASE –EVENT —*GAMES —!CONFERENCE —*PHENOMENA —*WAR —*NATURAL_DISASTER —*CRIME –TITLE —!POSITION_TITLE –*LANGUAGE –*RELIGION -NATURAL_OBJECT –ANIMAL –VEGETABLE –MINERAL -COLOR -TIME_TOP –TIMEX —TIME —DATE —*ERA –PERIODX —*TIME_PERIOD —*DATE_PERIOD —*WEEK_PERIOD —*MONTH_PERIOD —!YEAR_PERIOD -NUMEX –MONEY –*STOCK_INDEX –*POINT –PERCENT –MULTIPLICATION –FREQUENCY –*RANK –AGE –MEASUREMENT —PHYSICAL_EXTENT —SPACE —VOLUME —WEIGHT —*SPEED
		Continued on next page

Author(s)	Total types	Type taxonomy
		—*INTENSITY —*TEMPERATURE —*CALORIE —*SEISMIC_INTENSITY –COUNTX —N_PERSON —N_ORGANIZATION —N_LOCATION —-*N_COUNTRY —*N_FACILITY —N_PRODUCT —*N_EVENT —*N_ANIMAL —*N_VEGETABLE —*N_MINERAL *OTHER

Bibliography

Abney, S. (1991). Parsing by chunks, *in* S. P. Abney, R. C. Berwick & C. Tenny (eds), *Principle-based Parsing: Computation and Psycholinguistics*, Kluwer, pp. 257–278.

Androutsopoulos, I., Ritchie, G. D. & Thanisch, P. (1995). Natural language interafces to databases - an introduction, *Journal of Language Engineering* **1**(1): 29–81.

Breck, E., Burger, J., Ferro, L., House, D., Light, M. & Mani, I. (1999). A sys called Qanda, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Carbonell, J., Harman, D., Hovy, E., Maiorano, S., Prange, J. & Sparck-Jones, K. (2000). Vision statement to guide research in Question & Answering (Q&A) and text summarization, *Technical report*, NIST.
URL:
http://www-nlpir.nist.gov/projects/duc/papers/Final-Vision-Paper-v1a.pdf

Chang, C.-C. & Lin, C.-L. (2001). *LIBSVM: a Library for Support Vector Machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Charniak, E. (2000). A maximum-entropy-inspired parser, *1st Conference of the North American Chapter of the Association for Computational Linguistics and 6th Conference on Applied Natural Language Processing (ANLP/NAACL 2000)*.

Chinchor, N. (1997). MUC-7 named entity task definition, *Message Understanding Conference Proceedings (MUC-7)*.

Cohen, W. W. (1996). Learning trees and rules with set-valued features, *Proceedings of the thirteenth National Conference on Artificial Intelligence (AAAI-96)*.

Collins, M. & Duffy, N. (2001). Convolution kernels for natural language, *Proceedings of Neural Information Processing Systems (NIPS14)*.

Cortes, C. & Vapnik, V. N. (1995). Support vector machines, *Machine Learning* **20**: 273–297.

Dumais, S. T., Platt, J., Hechkerman, D. & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization, *Proceedings of ACM-CIKM98*, pp. 148–155.

Eichmann, D. & Srinivasan, P. (1999). Filters, webs and answers: The university of Iowa TREC-8 results, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*, MIT Press.

Ferret, F., Grau, B., Illouz, G., Jacquemin, C. & Masson, N. (1999). QALC - the question-answering program of the language and cognition group at LIMSI-CNRS, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Green, B., Wolf, A., Chomsky, C. & Laughery, K. (1961). BASEBALL: An automatic question answerer, *Proceedings of the Western Joint Computer Conference*, pp. 219–224.

Hacioglu, K. & Ward, W. (2003). Question classification with support vector machines and error correcting codes, *Proceedings of HLT-NACCL 2003*.

Harabagiu, S. M., Maiorano, S. J. & Paşca, M. A. (2003). Open-domain textual question answering techniques, *Natural Language Engineering* **9**(3): 231–267.

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V. & Morărescu, P. (2000). FALCON: Boosting knowledge for answer engines, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Harabagiu, S., Moldovan, D., Paşca, M., Surdeanu, M., Mihalcea, R., Gîrju, R., Rus, V., Lăcătuşu, F., Morărescu, P. & Bunescu, R. (2001). Answering complex, list and context questions with LCC's question-answering server, *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*.

Hermjakob, U. (2001). Parsing and question classification for question answering, *ACL-2001 Workshop on Open-Domain Question Answering*.

Hermjakob, U. & Mooney, R. J. (1997). Learning parse and translation decisions from examples with rich context, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 482–489.

Hirschman, L. & Gaizauskas, R. (2001). Natural language question answering: the view from here, *Natural Language Engineering* **7**(4): 275–300.

Hirschman, L., Light, M., Breck, E. & Burger, J. D. (1999). Deep Read: a reading comprehension system, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 325–332.

Hovy, E., Gerber, L., Hermjakob, U., Lin, C. & Ravichandran, D. (2001). Toward semantics-based answer pinpointing, *Proceedings of the DARPA Human Language Technology Conference (HLT)*, San Diego, CA.

Hovy, E. H., Hermjakob, U. & Ravichandran, D. (2002). A question/answer typology with surface text patterns, *Proceedings of the Human Language Technology (HLT) Conference*.

Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks* **13**: 415–425.

Hull, D. A. (1999). Xerox TREC-8 question answering track report, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Humphreys, K., Gaizauskas, R., Hepplea, M. & Sanderson, M. b. (1999). University of Sheffield TREC-8 Q & A system, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Ittycheriah, A., Franz, M., Zhu, W.-J. & Ratnaparkhi, A. (2000). IBM's statistical question answering system, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, *Proceedings of ECML98, 10th Eurpopean Conference on Machine Learning*.

Kazawa, H., Isozaki, H. & Maeda, E. (2001). NTT question answering system in TREC 2001, *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*.

Kim, S.-M., Baek, D.-H., Kim, S.-B. & Rim, H.-C. (2000). Question answering considering semantic categories and co-occurence density, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Laszlo, M., Kosseim, L. & Lapalme, G. (1999). Goal-driven answer extraction, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Li, X. & Roth, D. (2002). Learning question classifiers, *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 556–562.

Litkowski, K. (1999). Question-answering using semantic relation triples, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Mahesh, K. & Nirenburg, S. (1995). A situated ontology for practical NLP, *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligences (IJCAI-95)*.

Mitchell, T. M. (1997). *Machine Learning*, The McGraw-Hill Companies, Inc.

Moldovan, D., Harabagiu, S., Paşca, M., Mihalcea, R., Goodrum, R., Gîrju, R. & V., R. (1999). LASSO: A tool for surfing the answer net, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Moldovan, D., Paşca, M., Harabagiu, S. & Surdeanu, M. (2002). Performance issues and error analysis in an open-domain question answering system, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, pp. 33–40.

Oard, D. W., Wang, J., Lin, D. & Soboroff, I. (1999). TREC-8 experiments at Maryland: CLIR, QA and routing, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Ogden, B., Cowie, J., Ludovik, E., Molina-Salgado, H., Nirenburg, S., Sharples, N. & Sheremtyeva, S. (1999). CRL's TREC-8 systems cross-lingual IR, and Q&A, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Paşca, M. (2003). *Open-domain Question Answering from Large Text Collections*, CSLI Publications.

Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W. & Wei, X. (2002). QuASM: A system for question answering using semi-structured data, *Proceedings of the Joint Conference on Digital Libraries 2002*.

Prager, J., Chu-Carroll, J., Czuba, K., Welty, C., Ittycheriah, A. & Mahindru, R. (2003). IBM's PIQUANT in TREC2003, *NIST Special Publication: SP 500-255 The Twelfth Text Retrieval Conference (TREC 2003)*.

Prager, J., Radev, D., Brown, E., Coden, A. & Samn, V. (1999). The use of predictive annotation for question answering in TREC8, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Radev, D., Fan, W., Qi, H., Wu, H. & Grewal, A. (2002). Probabilistic question answering on the web, *Proceedings of the eleventh international conference on World Wide Web (WWW2002)*, Hawaii.

Rendell, L. & Cho, H. (1990). Empirical learning as a function of concept character, *Machine Learning* **5**(3): 267–298.

Rosenfeld, R. (2000). Two decades of statistical language modelling : Where do we go from here, *Proceedings of the IEEE*, Vol. 88, pp. 1270–1278.

Roth, D. (1998). Learning to resolve natural language ambiguities: a unified approach, *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, Madison, US, pp. 806–813.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys* **34**(1): 1–47.

Simmons, R. F. (1965). Answering english questions by computer: a survey, *Communications of the ACM* **8**(1): 53–70.

Singhal, A., Abney, S., Bacchiani, M., Collins, M., Hindle, D. & Pereira, F. (1999). AT&T at TREC-8, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Srihari, R. & Li, W. (1999). Information extraction supported question answering, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*.

Suzuki, J., Hirao, T., Sasaki, Y. & Maeda, E. (2003). Hierarchical directed acyclic graph kernel: Methods for natural language data, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL2003)*, pp. 32–29.

Suzuki, J., Sasaki, Y. & Maeda, E. (2002). SVM answer selection for open-domain question answering, *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*.

Suzuki, J., Taira, H., Sasaki, Y. & Maeda, E. (2003). Question classification using HDAG kernel, *The ACL 2003 Workshop on Multilingual Summarization and Question Answering*.

Takaki, T. (2000). NTT DATA TREC-9 question answering track report, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Voorhees, E. M. (2001). The TREC question answering track, *Natural Language Engineering* **7**(4): 361–378.

Witten, I. H. & Frank, E. (2000). *Data Mining: Practical Machine Learning Tools with Java Implementations*, Morgan Kaufmann.

Wu, L., Huang, X.-J., Guo, Y., Liu, B. & Zhang, Y. (2000). FDU at TREC9: CLIR, filtering and QA tasks, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*.

Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, pp. 42–49.

Zeng, Z. (2002). Answerbus question answering system, *Human Language Technology Conference (HLT 2002)*, San Diego, CA.

Zhang, D. & Lee, W. S. (2003a). A language modeling approach to passage question answering, *NIST Special Publication: SP 500-255 The Twelfth Text Retrieval Conference (TREC 2003)*.

Zhang, D. & Lee, W. S. (2003b). Question classification using support vector machines, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 26–32.

LINKÖPINGS UNIVERSITET

Avdelning, institution
Division, department

Institutionen för datavetenskap

Department of Computer and Information Science

Datum
Date

2007-06-18

Språk
Language

[ ] Svenska/Swedish
[X] Engelska/English
[ ] _____

Rapporttyp
Report category

[X] Licentiatavhandling
[ ] Examensarbete
[ ] C-uppsats
[ ] D-uppsats
[ ] Övrig rapport

_____

URL för elektronisk version

ISBN 978-91-85831-55-5

ISRN LiU-Tek-Lic-2007:29

Serietitel och serienummer
Title of series, numbering

ISSN 0280-7971

Linköping Studies in Science and Technology

Thesis No. 1320

Titel

Question Classification in Question Answering Systems

Författare

Håkan Sundblad

Sammanfattning

Question answering systems can be seen as the next step in information retrieval, allowing users to pose questions in natural language and receive succinct answers. In order for a question answering system as a whole to be successful, research has shown that the correct classification of questions with regards to the expected answer type is imperative. Question classification has two components: a taxonomy of answer types, and a machinery for making the classifications.

This thesis focuses on five different machine learning algorithms for the question classification task. The algorithms are k nearest neighbours, naïve bayes, decision tree learning, sparse network of winnows, and support vector machines. These algorithms have been applied to two different corpora, one of which has been used extensively in previous work and has been constructed for a specific agenda. The other corpus is drawn from a set of users' questions posed to a running online system. The results showed that the performance of the algorithms on the different corpora differs both in absolute terms, as well as with regards to the relative ranking of them. On the novel corpus, naïve bayes, decision tree learning, and support vector machines perform on par with each other, while on the biased corpus there is a clear difference between them, with support vector machines being the best and naïve bayes being the worst.

The thesis also presents an analysis of questions that are problematic for all learning algorithms. The errors can roughly be divided as due to categories with few members, variations in question formulation, the actual usage of the taxonomy, keyword errors, and spelling errors. A large portion of the errors were also hard to explain.

Nyckelord
Keywords
Question Classification, Question Answering Systems, Machine Learning, Taxonomy

Department of Computer and Information Science
Linköpings universitet

Linköping Studies in Science and Technology
Faculty of Arts and Sciences - Licentiate Theses

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. (Available at: FOA, Box 1165, S-581 11 Linköping, Sweden. FOA Report B30062E)

No 28 **Arne Jönsson, Mikael Patel:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms, 1984.

No 29 **Johnny Eckerland:** Retargeting of an Incremental Code Generator, 1984.

No 48 **Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching, 1985.

No 52 **Zebo Peng:** Steps Towards the Formalization of Designing VLSI Systems, 1985.

No 60 **Johan Fagerström:** Simulation and Evaluation of Architecture based on Asynchronous Processes, 1985.

No 71 **Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System, 1987.

No 72 **Tony Larsson:** On the Specification and Verification of VLSI Systems, 1986.

No 73 **Ola Strömfors:** A Structure Editor for Documents and Programs, 1986.

No 74 **Christos Levcopoulos:** New Results about the Approximation Behavior of the Greedy Triangulation, 1986.

No 104 **Shamsul I. Chowdhury: S**tatistical Expert Systems - a Special Application Area for Knowledge-Based Computer Methodology, 1987.

No 108 **Rober Bilos:** Incremental Scanning and Token-Based Editing, 1987.

No 111 **Hans Block:** SPORT-SORT Sorting Algorithms and Sport Tournaments, 1987.

No 113 **Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge, 1987.

No 118 **Mariam Kamkar, Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs, 1987.

No 126 **Dan Strömberg:** Transfer and Distribution of Application Programs, 1987.

No 127 **Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert Systems, 1987.

No 139 **Christer Bäckström:** Reasoning about Interdependent Actions, 1988.

No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing, 1988.

No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System, 1988.

No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.

No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.

No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models, 1989.

No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment, 1989.

No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager, 1989.

No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition, 1989.

No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems, 1989.

No 187 **Simin Nadjm-Tehrani:** Contributions to the Declarative Approach to Debugging Prolog Programs, 1989.

No 189 **Magnus Merkel:** Temporal Information in Natural Language, 1989.

No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs, 1989.

No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework, 1989.

No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action, 1990.

No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis, 1990.

No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning, 1990.

No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring, 1990.

No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems, 1990.

No 253 **Torbjörn Näslund:** SLDFA-Resolution - Computing Answers for Negative Queries, 1990.

No 260 **Peter D. Holmes:** Using Connectivity Graphs to Support Map-Related Reasoning, 1991.

No 283 **Olof Johansson:** Improving Implementation of Graphical User Interfaces for Object-Oriented Knowledge-Bases, 1991.

No 298 **Rolf G Larsson:** Aktivitetsbaserad kalkylering i ett nytt ekonomisystem, 1991.

No 318 **Lena Strömbäck:** Studies in Extended Unification-Based Formalism for Linguistic Description: An Algorithm for Feature Structures with Disjunction and a Proposal for Flexible Systems, 1992.

No 319 **Mikael Pettersson:** DML-A Language and System for the Generation of Efficient Compilers from Denotational Specification, 1992.

No 326 **Andreas Kågedal:** Logic Programming with External Procedures: an Implementation, 1992.

No 328 **Patrick Lambrix:** Aspects of Version Management of Composite Objects, 1992.

No 333 **Xinli Gu:** Testability Analysis and Improvement in High-Level Synthesis Systems, 1992.

No 335 **Torbjörn Näslund:** On the Role of Evaluations in Iterative Development of Managerial Support Sytems, 1992.

No 348 **Ulf Cederling:** Industrial Software Development - a Case Study, 1992.

No 352 **Magnus Morin:** Predictable Cyclic Computations in Autonomous Systems: A Computational Model and Implementation, 1992.

No 371 **Mehran Noghabai:** Evaluation of Strategic Investments in Information Technology, 1993.

No 378 **Mats Larsson:** A Transformational Approach to Formal Digital System Design, 1993.

No 380 **Johan Ringström:** Compiler Generation for Parallel Languages from Denotational Specifications, 1993.

No 381 **Michael Jansson:** Propagation of Change in an Intelligent Information System, 1993.

No 383 **Jonni Harrius:** An Architecture and a Knowledge Representation Model for Expert Critiquing Systems, 1993.

No 386 **Per Österling:** Symbolic Modelling of the Dynamic Environments of Autonomous Agents, 1993.

No 398 **Johan Boye:** Dependency-based Groudness Analysis of Functional Logic Programs, 1993.

No 402 **Lars Degerstedt:** Tabulated Resolution for Well Founded Semantics, 1993.
No 406 **Anna Moberg:** Satellitkontor - en studie av kommunikationsmönster vid arbete på distans, 1993.
No 414 **Peter Carlsson:** Separation av företagsledning och finansiering - fallstudier av företagsledarutköp ur ett agentteoretiskt perspektiv, 1994.
No 417 **Camilla Sjöström:** Revision och lagreglering - ett historiskt perspektiv, 1994.
No 436 **Cecilia Sjöberg:** Voices in Design: Argumentation in Participatory Development, 1994.
No 437 **Lars Viklund:** Contributions to a High-level Programming Environment for a Scientific Computing, 1994.
No 440 **Peter Loborg:** Error Recovery Support in Manufacturing Control Systems, 1994.
FHS 3/94 **Owen Eriksson:** Informationssystem med verksamhetskvalitet - utvärdering baserat på ett verksamhetsinriktat och samskapande perspektiv, 1994.
FHS 4/94 **Karin Pettersson:** Informationssystemstrukturering, ansvarsfördelning och användarinflytande - En komparativ studie med utgångspunkt i två informationssystemstrategier, 1994.
No 441 **Lars Poignant:** Informationsteknologi och företagsetablering - Effekter på produktivitet och region, 1994.
No 446 **Gustav Fahl:** Object Views of Relational Data in Multidatabase Systems, 1994.
No 450 **Henrik Nilsson:** A Declarative Approach to Debugging for Lazy Functional Languages, 1994.
No 451 **Jonas Lind:** Creditor - Firm Relations: an Interdisciplinary Analysis, 1994.
No 452 **Martin Sköld:** Active Rules based on Object Relational Queries - Efficient Change Monitoring Techniques, 1994.
No 455 **Pär Carlshamre:** A Collaborative Approach to Usability Engineering: Technical Communicators and System Developers in Usability-Oriented Systems Development, 1994.
FHS 5/94 **Stefan Cronholm:** Varför CASE-verktyg i systemutveckling? - En motiv- och konsekvensstudie avseende arbetssätt och arbetsformer, 1994.
No 462 **Mikael Lindvall:** A Study of Traceability in Object-Oriented Systems Development, 1994.
No 463 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av Sandviks förvärv av Bahco Verktyg, 1994.
No 464 **Hans Olsén:** Collage Induction: Proving Properties of Logic Programs by Program Synthesis, 1994.
No 469 **Lars Karlsson:** Specification and Synthesis of Plans Using the Features and Fluents Framework, 1995.
No 473 **Ulf Söderman:** On Conceptual Modelling of Mode Switching Systems, 1995.
No 475 **Choong-ho Yi:** Reasoning about Concurrent Actions in the Trajectory Semantics, 1995.
No 476 **Bo Lagerström:** Successiv resultatavräkning av pågående arbeten. - Fallstudier i tre byggföretag, 1995.
No 478 **Peter Jonsson:** Complexity of State-Variable Planning under Structural Restrictions, 1995.
FHS 7/95 **Anders Avdic:** Arbetsintegrerad systemutveckling med kalkylkprogram, 1995.
No 482 **Eva L Ragnemalm:** Towards Student Modelling through Collaborative Dialogue with a Learning Companion, 1995.
No 488 **Eva Toller:** Contributions to Parallel Multiparadigm Languages: Combining Object-Oriented and Rule-Based Programming, 1995.
No 489 **Erik Stoy:** A Petri Net Based Unified Representation for Hardware/Software Co-Design, 1995.
No 497 **Johan Herber:** Environment Support for Building Structured Mathematical Models, 1995.
No 498 **Stefan Svenberg:** Structure-Driven Derivation of Inter-Lingual Functor-Argument Trees for Multi-Lingual Generation, 1995.
No 503 **Hee-Cheol Kim:** Prediction and Postdiction under Uncertainty, 1995.
FHS 8/95 **Dan Fristedt:** Metoder i användning - mot förbättring av systemutveckling genom situationell metodkunskap och metodanalys, 1995.
FHS 9/95 **Malin Bergvall:** Systemförvaltning i praktiken - en kvalitativ studie avseende centrala begrepp, aktiviteter och ansvarsroller, 1995.
No 513 **Joachim Karlsson:** Towards a Strategy for Software Requirements Selection, 1995.
No 517 **Jakob Axelsson:** Schedulability-Driven Partitioning of Heterogeneous Real-Time Systems, 1995.
No 518 **Göran Forslund:** Toward Cooperative Advice-Giving Systems: The Expert Systems Experience, 1995.
No 522 **Jörgen Andersson:** Bilder av småföretagares ekonomistyrning, 1995.
No 538 **Staffan Flodin:** Efficient Management of Object-Oriented Queries with Late Binding, 1996.
No 545 **Vadim Engelson:** An Approach to Automatic Construction of Graphical User Interfaces for Applications in Scientific Computing, 1996.
No 546 **Magnus Werner :** Multidatabase Integration using Polymorphic Queries and Views, 1996.
FiF-a 1/96 **Mikael Lind:** Affärsprocessinriktad förändringsanalys - utveckling och tillämpning av synsätt och metod, 1996.
No 549 **Jonas Hallberg:** High-Level Synthesis under Local Timing Constraints, 1996.
No 550 **Kristina Larsen:** Förutsättningar och begränsningar för arbete på distans - erfarenheter från fyra svenska företag. 1996.
No 557 **Mikael Johansson:** Quality Functions for Requirements Engineering Methods, 1996.
No 558 **Patrik Nordling:** The Simulation of Rolling Bearing Dynamics on Parallel Computers, 1996.
No 561 **Anders Ekman:** Exploration of Polygonal Environments, 1996.
No 563 **Niclas Andersson:** Compilation of Mathematical Models to Parallel Code, 1996.
No 567 **Johan Jenvald:** Simulation and Data Collection in Battle Training, 1996.
No 575 **Niclas Ohlsson:** Software Quality Engineering by Early Identification of Fault-Prone Modules, 1996.
No 576 **Mikael Ericsson:** Commenting Systems as Design Support—A Wizard-of-Oz Study, 1996.
No 587 **Jörgen Lindström:** Chefers användning av kommunikationsteknik, 1996.
No 589 **Esa Falkenroth:** Data Management in Control Applications - A Proposal Based on Active Database Systems, 1996.
No 591 **Niclas Wahllöf:** A Default Extension to Description Logics and its Applications, 1996.
No 595 **Annika Larsson:** Ekonomisk Styrning och Organisatorisk Passion - ett interaktivt perspektiv, 1997.
No 597 **Ling Lin:** A Value-based Indexing Technique for Time Sequences, 1997.

No 598 **Rego Granlund:** C^3Fire - A Microworld Supporting Emergency Management Training, 1997.
No 599 **Peter Ingels:** A Robust Text Processing Technique Applied to Lexical Error Recovery, 1997.
No 607 **Per-Arne Persson:** Toward a Grounded Theory for Support of Command and Control in Military Coalitions, 1997.
No 609 **Jonas S Karlsson:** A Scalable Data Structure for a Parallel Data Server, 1997.
FiF-a 4 **Carita Åbom:** Videomötesteknik i olika affärssituationer - möjligheter och hinder, 1997.
FiF-a 6 **Tommy Wedlund**: Att skapa en företagsanpassad systemutvecklingsmodell - genom rekonstruktion, värdering och vidareutveckling i T50-bolag inom ABB, 1997.
No 615 **Silvia Coradeschi**: A Decision-Mechanism for Reactive and Coordinated Agents, 1997.
No 623 **Jan Ollinen:** Det flexibla kontorets utveckling på Digital - Ett stöd för multiflex? 1997.
No 626 **David Byers:** Towards Estimating Software Testability Using Static Analysis, 1997.
No 627 **Fredrik Eklund:** Declarative Error Diagnosis of GAPLog Programs, 1997.
No 629 **Gunilla Ivefors:** Krigsspel coh Informationsteknik inför en oförutsägbar framtid, 1997**.**
No 631 **Jens-Olof Lindh:** Analysing Traffic Safety from a Case-Based Reasoning Perspective, 1997
No 639 **Jukka Mäki-Turja:**. Smalltalk - a suitable Real-Time Language, 1997.
No 640 **Juha Takkinen**: CAFE: Towards a Conceptual Model for Information Management in Electronic Mail, 1997.
No 643 **Man Lin**: Formal Analysis of Reactive Rule-based Programs, 1997.
No 653 **Mats Gustafsson**: Bringing Role-Based Access Control to Distributed Systems, 1997.
FiF-a 13 **Boris Karlsson:** Metodanalys för förståelse och utveckling av systemutvecklingsverksamhet. Analys och värdering av systemutvecklingsmodeller och dess användning, 1997.
No 674 **Marcus Bjäreland:** Two Aspects of Automating Logics of Action and Change - Regression and Tractability, 1998.
No 676 **Jan Håkegård**: Hiera rchical Test Architecture and Board-Level Test Controller Synthesis, 1998.
No 668 **Per-Ove Zetterlund**: Normering av svensk redovisning - En studie av tillkomsten av Redovisningsrådets rekommendation om koncernredovisning (RR01:91), 1998.
No 675 **Jimmy Tjäder**: Projektledaren & planen - en studie av projektledning i tre installations- och systemutvecklingsprojekt, 1998.
FiF-a 14 **Ulf Melin**: Informationssystem vid ökad affärs- och processorientering - egenskaper, strategier och utveckling, 1998.
No 695 **Tim Heyer**: COMPASS: Introduction of Formal Methods in Code Development and Inspection, 1998.
No 700 **Patrik Hägglund:** Programming Languages for Computer Algebra, 1998.
FiF-a 16 **Marie-Therese Christiansson:** Inter-organistorisk verksamhetsutveckling - metoder som stöd vid utveckling av partnerskap och informationssystem, 1998.
No 712 **Christina Wennestam:** Information om immateriella resurser. Investeringar i forskning och utveckling samt i personal inom skogsindustrin, 1998.
No 719 **Joakim Gustafsson:** Extending Temporal Action Logic for Ramification and Concurrency, 1998.
No 723 **Henrik André-Jönsson:** Indexing time-series data using text indexing methods, 1999.
No 725 **Erik Larsson:** High-Level Testability Analysis and Enhancement Techniques, 1998.
No 730 **Carl-Johan Westin:** Informationsförsörjning: en fråga om ansvar - aktiviteter och uppdrag i fem stora svenska organisationers operativa informationsförsörjning, 1998.
No 731 **Åse Jansson:** Miljöhänsyn - en del i företags styrning, 1998.
No 733 **Thomas Padron-McCarthy:** Performance-Polymorphic Declarative Queries, 1998.
No 734 **Anders Bäckström:** Värdeskapande kreditgivning - Kreditriskhantering ur ett agentteoretiskt perspektiv, 1998.
FiF-a 21 **Ulf Seigerroth:** Integration av förändringsmetoder - en modell för välgrundad metodintegration, 1999.
FiF-a 22 **Fredrik Öberg:** Object-Oriented Frameworks - A New Strategy for Case Tool Development, 1998.
No 737 **Jonas Mellin:** Predictable Event Monitoring, 1998.
No 738 **Joakim Eriksson:** Specifying and Managing Rules in an Active Real-Time Database System, 1998.
FiF-a 25 **Bengt E W Andersson:** Samverkande informationssystem mellan aktörer i offentliga åtaganden - En teori om aktörsarenor i samverkan om utbyte av information, 1998.
No 742 **Pawel Pietrzak:** Static Incorrectness Diagnosis of CLP (FD), 1999.
No 748 **Tobias Ritzau:** Real-Time Reference Counting in RT-Java, 1999.
No 751 **Anders Ferntoft:** Elektronisk affärskommunikation - kontaktkostnader och kontaktprocesser mellan kunder och leverantörer på producentmarknader,1999.
No 752 **Jo Skåmedal:** Arbete på distans och arbetsformens påverkan på resor och resmönster, 1999.
No 753 **Johan Alvehus:** Mötets metaforer. En studie av berättelser om möten, 1999.
No 754 **Magnus Lindahl:** Bankens villkor i låneavtal vid kreditgivning till högt belånade företagsförvärv: En studie ur ett agentteoretiskt perspektiv, 2000.
No 766 **Martin V. Howard:** Designing dynamic visualizations of temporal data, 1999.
No 769 **Jesper Andersson:** Towards Reactive Software Architectures, 1999.
No 775 **Anders Henriksson:** Unique kernel diagnosis, 1999.
FiF-a 30 **Pär J. Ågerfalk:** Pragmatization of Information Systems - A Theoretical and Methodological Outline, 1999.
No 787 **Charlotte Björkegren:** Learning for the next project - Bearers and barriers in knowledge transfer within an organisation, 1999.
No 788 **Håkan Nilsson:** Informationsteknik som drivkraft i granskningsprocessen - En studie av fyra revisionsbyråer, 2000.
No 790 **Erik Berglund:** Use-Oriented Documentation in Software Development, 1999.
No 791 **Klas Gäre:** Verksamhetsförändringar i samband med IS-införande, 1999.
No 800 **Anders Subotic:** Software Quality Inspection, 1999.
No 807 **Svein Bergum**: Managerial communication in telework, 2000.

No 809 **Flavius Gruian:** Energy-Aware Design of Digital Systems, 2000.
FiF-a 32 **Karin Hedström:** Kunskapsanvändning och kunskapsutveckling hos verksamhetskonsulter - Erfarenheter från ett FOU-samarbete, 2000.
No 808 **Linda Askenäs:** Affärssystemet - En studie om teknikens aktiva och passiva roll i en organisation, 2000.
No 820 **Jean Paul Meynard:** Control of industrial robots through high-level task programming, 2000.
No 823 **Lars Hult:** Publika Gränsytor - ett designexempel, 2000.
No 832 **Paul Pop:** Scheduling and Communication Synthesis for Distributed Real-Time Systems, 2000.
FiF-a 34 **Göran Hultgren:** Nätverksinriktad Förändringsanalys - perspektiv och metoder som stöd för förståelse och utveckling av affärsrelationer och informationssystem, 2000.
No 842 **Magnus Kald:** The role of management control systems in strategic business units, 2000.
No 844 **Mikael Cäker:** Vad kostar kunden? Modeller för intern redovisning, 2000.
FiF-a 37 **Ewa Braf**: Organisationers kunskapsverksamheter - en kritisk studie av ”knowledge management”, 2000.
FiF-a 40 **Henrik Lindberg:** Webbaserade affärsprocesser - Möjligheter och begränsningar, 2000.
FiF-a 41 **Benneth Christiansson:** Att komponentbasera informationssystem - Vad säger teori och praktik?, 2000.
No. 854 **Ola Pettersson:** Deliberation in a Mobile Robot, 2000.
No 863 **Dan Lawesson**: Towards Behavioral Model Fault Isolation for Object Oriented Control Systems, 2000.
No 881 **Johan Moe:** Execution Tracing of Large Distributed Systems, 2001.
No 882 **Yuxiao Zhao:** XML-based Frameworks for Internet Commerce and an Implementation of B2B e-procurement, 2001.
No 890 **Annika Flycht-Eriksson:** Domain Knowledge Management inInformation-providing Dialogue systems, 2001.
FiF-a 47 **Per-Arne Segerkvis**t: Webbaserade imaginära organisationers samverkansformer: Informationssystemarkitektur och aktörssamverkan som förutsättningar för affärsprocesser, 2001.
No 894 **Stefan Svarén:** Styrning av investeringar i divisionaliserade företag - Ett koncernperspektiv, 2001.
No 906 **Lin Han**: Secure and Scalable E-Service Software Delivery, 2001.
No 917 **Emma Hansson:** Optionsprogram för anställda - en studie av svenska börsföretag, 2001.
No 916 **Susanne Odar:** IT som stöd för strategiska beslut, en studie av datorimplementerade modeller av verksamhet som stöd för beslut om anskaffning av JAS 1982, 2002.
FiF-a-49 **Stefan Holgersson:** IT-system och filtrering av verksamhetskunskap - kvalitetsproblem vid analyser och beslutsfattande som bygger på uppgifter hämtade från polisens IT-system, 2001.
FiF-a-51 **Per Oscarsson:**Informationssäkerhet i verksamheter - begrepp och modeller som stöd för förståelse av informationssäkerhet och dess hantering, 2001.
No 919 **Luis Alejandro Cortes:** A Petri Net Based Modeling and Verification Technique for Real-Time Embedded Systems, 2001.
No 915 **Niklas Sandell:** Redovisning i skuggan av en bankkris - Värdering av fastigheter. 2001.
No 931 **Fredrik Elg:** Ett dynamiskt perspektiv på individuella skillnader av heuristisk kompetens, intelligens, mentala modeller, mål och konfidens i kontroll av mikrovärlden Moro, 2002.
No 933 **Peter Aronsson:** Automatic Parallelization of Simulation Code from Equation Based Simulation Languages, 2002.
No 938 **Bourhane Kadmiry**: Fuzzy Control of Unmanned Helicopter, 2002.
No 942 **Patrik Haslum**: Prediction as a Knowledge Representation Problem: A Case Study in Model Design, 2002.
No 956 **Robert Sevenius:** On the instruments of governance - A law & economics study of capital instruments in limited liability companies, 2002.
FiF-a 58 **Johan Petersson:** Lokala elektroniska marknadsplatser - informationssystem för platsbundna affärer, 2002.
No 964 **Peter Bunus:** Debugging and Structural Analysis of Declarative Equation-Based Languages, 2002.
No 973 **Gert Jervan:** High-Level Test Generation and Built-In Self-Test Techniques for Digital Systems, 2002.
No 958 **Fredrika Berglund:** Management Control and Strategy - a Case Study of Pharmaceutical Drug Development, 2002.
FiF-a 61 **Fredrik Karlsson:** Meta-Method for Method Configuration - A Rational Unified Process Case, 2002.
No 985 **Sorin Manolache:** Schedulability Analysis of Real-Time Systems with Stochastic Task Execution Times, 2002.
No 982 **Diana Szentiványi:** Performance and Availability Trade-offs in Fault-Tolerant Middleware, 2002.
No 989 **Iakov Nakhimovski:** Modeling and Simulation of Contacting Flexible Bodies in Multibody Systems, 2002.
No 990 **Levon Saldamli:** PDEModelica - Towards a High-Level Language for Modeling with Partial Differential Equations, 2002.
No 991 **Almut Herzog:** Secure Execution Environment for Java Electronic Services, 2002.
No 999 **Jon Edvardsson:** Contributions to Program- and Specification-based Test Data Generation, 2002
No 1000 **Anders Arpteg:** Adaptive Semi-structured Information Extraction, 2002.
No 1001 **Andrzej Bednarski:** A Dynamic Programming Approach to Optimal Retargetable Code Generation for Irregular Architectures, 2002.
No 988 **Mattias Arvola:** Good to use! : Use quality of multi-user applications in the home, 2003.
FiF-a 62 **Lennart Ljung:** Utveckling av en projektivitetsmodell - om organisationers förmåga att tillämpa projektarbetsformen, 2003.
No 1003 **Pernilla Qvarfordt:** User experience of spoken feedback in multimodal interaction, 2003.
No 1005 **Alexander Siemers:** Visualization of Dynamic Multibody Simulation With Special Reference to Contacts, 2003.
No 1008 **Jens Gustavsson:** Towards Unanticipated Runtime Software Evolution, 2003.
No 1010 **Calin Curescu:** Adaptive QoS-aware Resource Allocation for Wireless Networks, 2003.
No 1015 **Anna Andersson:** Management Information Systems in Process-oriented Healthcare Organisations, 2003.
No 1018 **Björn Johansson:** Feedforward Control in Dynamic Situations, 2003.
No 1022 **Traian Pop:** Scheduling and Optimisation of Heterogeneous Time/Event-Triggered Distributed Embedded Systems, 2003.
FiF-a 65 **Britt-Marie Johansson:** Kundkommunikation på distans - en studie om kommunikationsmediets betydelse i affärstransaktioner, 2003.

No 1024 **Aleksandra Tešanovic:** Towards Aspectual Component-Based Real-Time System Development, 2003.
No 1034 **Arja Vainio-Larsson**: Designing for Use in a Future Context - Five Case Studies in Retrospect, 2003.
No 1033 **Peter Nilsson:** Svenska bankers redovisningsval vid reservering för befarade kreditförluster - En studie vid införandet av nya redovisningsregler, 2003.
FiF-a 69 **Fredrik Ericsson:** Information Technology for Learning and Acquiring of Work Knowledge, 2003.
No 1049 **Marcus Comstedt:** Towards Fine-Grained Binary Composition through Link Time Weaving, 2003.
No 1052 **Åsa Hedenskog:** Increasing the Automation of Radio Network Control, 2003.
No 1054 **Claudiu Duma:** Security and Efficiency Tradeoffs in Multicast Group Key Management, 2003.
FiF-a 71 **Emma Eliason:** Effektanalys av IT-systems handlingsutrymme, 2003.
No 1055 **Carl Cederberg:** Experiments in Indirect Fault Injection with Open Source and Industrial Software, 2003.
No 1058 **Daniel Karlsson:** Towards Formal Verification in a Component-based Reuse Methodology, 2003.
FiF-a 73 **Anders Hjalmarsson:** Att etablera och vidmakthålla förbättringsverksamhet - behovet av koordination och interaktion vid förändring av systemutvecklingsverksamheter, 2004.
No 1079 **Pontus Johansson:** Design and Development of Recommender Dialogue Systems, 2004.
No 1084 **Charlotte Stoltz:** Calling for Call Centres - A Study of Call Centre Locations in a Swedish Rural Region, 2004.
FiF-a 74 **Björn Johansson:** Deciding on Using Application Service Provision in SMEs, 2004.
No 1094 **Genevieve Gorrell:** Language Modelling and Error Handling in Spoken Dialogue Systems, 2004.
No 1095 **Ulf Johansson:** Rule Extraction - the Key to Accurate and Comprehensible Data Mining Models, 2004.
No 1099 **Sonia Sangari:** Computational Models of Some Communicative Head Movements, 2004.
No 1110 **Hans Nässla:** Intra-Family Information Flow and Prospects for Communication Systems, 2004.
No 1116 **Henrik Sällberg:** On the value of customer loyalty programs - A study of point programs and switching costs, 2004.
FiF-a 77 **Ulf Larsson:** Designarbete i dialog - karaktärisering av interaktionen mellan användare och utvecklare i en systemutvecklingsprocess, 2004.
No 1126 **Andreas Borg:** Contribution to Management and Validation of Non-Functional Requirements, 2004.
No 1127 **Per-Ola Kristensson:** Large Vocabulary Shorthand Writing on Stylus Keyboard, 2004.
No 1132 **Pär-Anders Albinsson:** Interacting with Command and Control Systems: Tools for Operators and Designers, 2004.
No 1130 **Ioan Chisalita:** Safety-Oriented Communication in Mobile Networks for Vehicles, 2004.
No 1138 **Thomas Gustafsson:** Maintaining Data Consistency im Embedded Databases for Vehicular Systems, 2004.
No 1149 **Vaida Jakoniené:** A Study in Integrating Multiple Biological Data Sources, 2005.
No 1156 **Abdil Rashid Mohamed**: High-Level Techniques for Built-In Self-Test Resources Optimization, 2005.
No 1162 **Adrian Pop:** Contributions to Meta-Modeling Tools and Methods, 2005.
No 1165 **Fidel Vascós Palacios:** On the information exchange between physicians and social insurance officers in the sick leave process: an Activity Theoretical perspective, 2005.
FiF-a 84 **Jenny Lagsten:** Verksamhetsutvecklande utvärdering i informationssystemprojekt, 2005.
No 1166 **Emma Larsdotter Nilsson:** Modeling, Simulation, and Visualization of Metabolic Pathways Using Modelica, 2005.
No 1167 **Christina Keller:** Virtual Learning Environments in higher education. A study of students' acceptance of educational technology, 2005.
No 1168 **Cécile Åberg:** Integration of organizational workflows and the Semantic Web, 2005.
FiF-a 85 **Anders Forsman:** Standardisering som grund för informationssamverkan och IT-tjänster - En fallstudie baserad på trafikinformationstjänsten RDS-TMC, 2005.
No 1171 **Yu-Hsing Huang:** A systemic traffic accident model, 2005.
FiF-a 86 **Jan Olausson:** Att modellera uppdrag - grunder för förståelse av processinriktade informationssystem i transaktionsintensiva verksamheter, 2005.
No 1172 **Petter Ahlström**: Affärsstrategier för seniorbostadsmarknaden, 2005.
No 1183 **Mathias Cöster**: Beyond IT and Productivity - How Digitization Transformed the Graphic Industry, 2005.
No 1184 **Åsa Horzella**: Beyond IT and Productivity - Effects of Digitized Information Flows in Grocery Distribution, 2005.
No 1185 **Maria Kollberg**: Beyond IT and Productivity - Effects of Digitized Information Flows in the Logging Industry, 2005.
No 1190 **David Dinka**: Role and Identity - Experience of technology in professional settings, 2005.
No 1191 **Andreas Hansson**: Increasing the Storage Capacity of Recursive Auto-associative Memory by Segmenting Data, 2005.
No 1192 **Nicklas Bergfeldt:** Towards Detached Communication for Robot Cooperation, 2005.
No 1194 **Dennis Maciuszek:** Towards Dependable Virtual Companions for Later Life, 2005.
No 1204 **Beatrice Alenljung:** Decision-making in the Requirements Engineering Process: A Human-centered Approach, 2005
No 1206 **Anders Larsson:** System-on-Chip Test Scheduling and Test Infrastructure Design, 2005.
No 1207 **John Wilander:** Policy and Implementation Assurance for Software Security, 2005.
No 1209 **Andreas Käll:** Översättningar av en managementmodell - En studie av införandet av Balanced Scorecard i ett landsting, 2005.
No 1225 **He Tan:** Aligning and Merging Biomedical Ontologies, 2006.
No 1228 **Artur Wilk:** Descriptive Types for XML Query Language Xcerpt, 2006.
No 1229 **Per Olof Pettersson:** Sampling-based Path Planning for an Autonomous Helicopter, 2006.
No 1231 **Kalle Burbeck:** Adaptive Real-time Anomaly Detection for Safeguarding Critical Networks, 2006.
No 1233 **Daniela Mihailescu:** Implementation Methodology in Action: A Study of an Enterprise Systems Implementation Methodology, 2006.
No 1244 **Jörgen Skågeby:** Public and Non-public gifting on the Internet, 2006.
No 1248 **Karolina Eliasson:** The Use of Case-Based Reasoning in a Human-Robot Dialog System, 2006.
No 1263 **Misook Park-Westman:** Managing Competence Development Programs in a Cross-Cultural Organisation-What are the Barriers and Enablers, 2006.
FiF-a 90 **Amra Halilovic:** Ett praktikperspektiv på hantering av mjukvarukomponenter, 2006.
No 1272 **Raquel Flodström**: A Framework for the Strategic Management of Information Technology, 2006.

No 1277 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Embedded Systems, 2006.
No 1283 **Håkan Hasewinkel:** A Blueprint for Using Commercial Games off the Shelf in Defence Training, Education and Research Simulations, 2006.
FiF-a 91 **Hanna Broberg:** Verksamhetsanpassade IT-stöd - Designteori och metod, 2006.
No 1286 **Robert Kaminski:** Towards an XML Document Restructuring Framework, 2006
No 1293 **Jiri Trnka:** Prerequisites for data sharing in emergency management, 2007.
No 1302 **Björn Hägglund:** A Framework for Designing Constraint Stores, 2007.
No 1303 **Daniel Andreasson**: Slack-Time Aware Dynamic Routing Schemes for On-Chip Networks, 2007.
No 1305 **Magnus Ingmarsson:** Modelling User Tasks and Intentions for Service Discovery in Ubiquitous Computing, 2007.
No 1306 **Gustaf Svedjemo**: Ontology as Conceptual Schema when Modelling Historical Maps for Database Storage, 2007.
No 1307 **Gianpaolo Conte**: Navigation Functionalities for an Autonomous UAV Helicopter, 2007.
No 1309 **Ola Leifler**: User-Centric Critiquing in Command and Control: The DKExpert and ComPlan Approaches, 2007.
No 1312 **Henrik Svensson**: Embodied simulation as off-line representation, 2007.
No 1313 **Zhiyuan He:** System-on-Chip Test Scheduling with Defect-Probability and Temperature Considerations, 2007.
No 1317 **Jonas Elmqvist:** Components, Safety Interfaces and Compositional Analysis, 2007.
No 1320 **Håkan Sundblad:** Question Classification in Question Answering Systems, 2007.