Analysis of Semantic Classes: Toward Non-Factoid Question Answering

by

Yun Niu

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Analysis of Semantic Classes: Toward Non-Factoid
Question Answering

Yun Niu
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2007

The task of question answering (QA) is to find the accurate and precise answer to a natural language question in some predefined text. Most existing QA systems handle fact-based questions that usually take named entities as the answers. In this thesis, we focus on a different type of QA—non-factoid QA (NFQA) to deal with more complex information needs. The goal of the present study is to propose approaches that tackle important problems in non-factoid QA.

We proposed an approach using semantic class analysis as the organizing principle to answer non-factoid questions. This approach contains four major components:

- Detecting semantic classes in questions and answer sources
- Identifying properties of semantic classes
- Question-answer matching: exploring properties of semantic classes to find relevant pieces of information
- Constructing answers by merging or synthesizing relevant information using relations between semantic classes

We investigated NFQA in the context of clinical question answering, and focused on three semantic classes that correspond to roles in the commonly accepted PICO format of describing clinical scenarios. The three classes are: the problem of the patient, the intervention used to treat the problem, and the clinical outcome.

We used rule-based approaches to identify clinical outcomes and relations between instances of interventions in sentences.

We identified an important property of semantic classes—their cores. We showed how cores of interventions, problems, and outcomes in a sentence can be extracted automatically by developing an approach exploring semi-supervised learning techniques. Another property that We analyzed is polarity, an inherent property of clinical outcomes. We developed a method using a supervised learning model to automatically detect polarity of clinical outcomes.

We built explicit connection between text summarization and identifying answer components in NFQA and constructed a summarization system that explores a supervised classification model to extract important sentences for answer construction. We investigated the role of clinical outcome and their polarity in this task.

Dedication

To my parents.

Acknowledgements

I wish to express my deep appreciation to my advisor, Graeme Hirst, for his help, which led me into the community of Computational Linguistics, and for his continued encouragement and support. His valuable guidance aided me immensely during the preparation of this thesis.

I would like to thank my committee members. I thank Gerald Penn and Suzanne Stevenson for helpful comments and suggestions that led to many improvements in the thesis. I thank John Mylopoulos and Sharon Straus for being in my thesis committee and their support in the EPoCare project, and Claire Cardie for being the external reviewer.

I would like to thank Xiaodan Zhu and Jianhua Li for their contributions, Xuming He for many helpful suggestions, and Afsaneh Fazly for her valuable comments on the writing of the thesis.

I would also like to thank Patricia Rodriguez Gianolli and Gregory McArthur for many interesting discussions on the EPoCare project.

I am grateful to have Afra Alishahi, Vivian Tsang, Saif Mohiuddin Mohammad, Faye Baron and other students in the Computational Linguistics group who made graduate study fun.

I am lucky to have many friends at the University of Toronto whose company and companionship are always there when I need it.

Finally, this thesis would not have been possible without the support and love of my mother, my father, and Ou. Thank you!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Question Answering?

As more and more information is accessible to users, more support from advanced technologies is required to help obtain the desired information. This brings new challenges to the area of information retrieval (IR) in both the query and the answer processing. To free the user from constructing a complicated boolean keywords query, the system should be able to process queries represented in natural language. Instead of replying with some documents relevant to the query, the system should answer the questions accurately and concisely. Systems with such characteristics are Question-Answering (QA) systems, which take advantage of high-quality natural language processing and mature technologies in IR. The task of a QA system is to find the answer to a particular natural language question in some predefined text.

Generally, current QA tasks can be classified into two categories: fact-based QA (FBQA) and non-factoid QA (NFQA). In FBQA, answers are usually named entities, such as *person name*, *time*, and *location*. For example:

Q: Who was the US president in 1999?

A: Bill Clinton

Q: Which city is the capital of China?

A: Beijing

NFQA aims to answer questions whose answers are not just named entities, such as questions

posed by clinicians in patient treatment:

Q: In a patient with a generalized anxiety disorder, does cognitive behaviour or relaxation therapy decrease symptoms?

Clinical outcomes of cognitive behaviour or relaxation therapy could be complicated. They could be beneficial or harmful; they could have different effects for different patient groups; some clinical trials may show they are beneficial while others don't. Answers to these questions can only be obtained by synthesizing relevant information.

Both FBQA and NFQA need to address some major research problems, and they fit into the same general QA framework. This thesis focuses on NFQA, and our working domain is medicine.

1.2 What are the new research problems posed by QA?

The first problem for QA is to understand the task. Since there are many different types of questions, it is very important for a QA system to know what a particular question is asking for. Some techniques have been recognized to be effective in FBQA, which are discussed in section 1.4.1 on the question-processing phase. In NFQA, however, it is much more difficult to understand the information needs.

Matching the answer to the question is another big challenge. Questions and the answers often have very different phrasings. Matching techniques need to find the correspondence between them. Compared to FBQA, such correspondence in NFQA is usually less explicit.

Answer generation is the last problem in QA. After the best candidates are selected by the matching techniques, they need to be processed to obtain accurate and concise answers.

1.3 Question-Answering framework

The architecture of a typical QA system is shown in Figure 1.1.

1. Question processing. The aim of the question processing is to understand the question. In most FBQA systems, this includes:

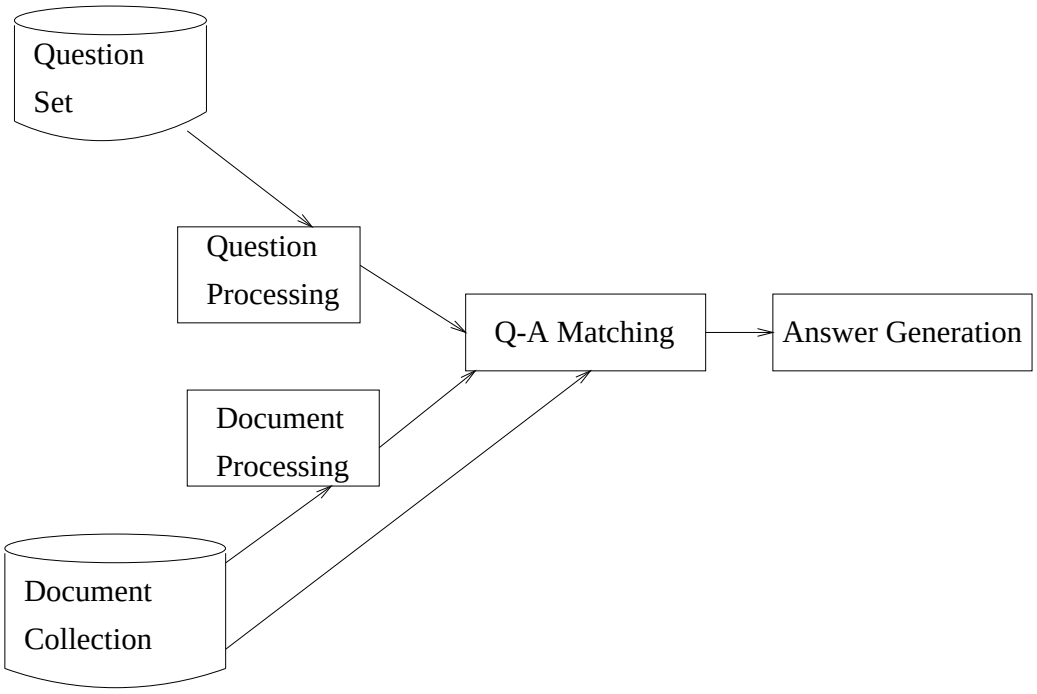


Figure 1.1: QA Architecture

- determining what type of question is being asked (e.g., where)
- inferring what kind of answer is expected (e.g., location)
- determining the focus of the question—its central point
- formulating a query on the document collection by using keywords in the question

Some NFQA systems suggest to clarify the question by communicating with users.

2. Document processing. Before the matching of question and answer starts, the documents in the collection may be transformed to some other representation so that efficient search can be performed. Many systems imported index technology from IR systems in this step.

3. Question-Answer matching. Before doing detailed analysis to find the answer, a relatively small set of candidates should be found. Conventional keyword matching and expected answer-type checking are often involved in this step. Unmatched candidates

will be filtered out directly.

To find the best answer, different techniques are used to analyze the relationship between the question and candidate answer thoroughly. Knowledge-intensive, data-intensive, and statistical approaches address the problem with different emphasis.

4. Answer generation. It has not been fully addressed in most current systems. Most systems just extract small fragments that contain the answer information from the answer candidates as the final answers. Even the extraction process is not discussed in detail in many works.

In the following two sections, related work in FBQA and NFQA is reviewed respectively to further understand their difference and connections, and state-of-the-art in QA.

1.4 Fact-based questions

The main problem in QA is the great variation in expressing the question and the answer. According to how it is addressed, current work in FBQA can be partitioned into two classes.

- Knowledge intensive. The intuition of the knowledge-intensive approaches is to find a proper meta-form so that both the question and the answer can be represented by it. The construction of the form usually exploits natural language processing technology as well as related real-world knowledge.

- Data intensive. The data-intensive approaches put the emphasis on prediction of the answer by using the evidence from the data set. For each question, some approaches try to compose all the possible answer formats and then compare them with the answer candidates to find the one that meets the prediction. Some approaches estimate how likely a candidate is the expected answer by collecting statistical data from a large candidate set.

The following two subsections will discuss some typical FBQA work in detail.

1.4.1 Knowledge-Intensive Approaches

The problems that are emphasized in knowledge-intensive systems are discussed in this section. For each problem, methods explored in different systems are compared.

Answer-type identification The type of the answer tells us the general category of the expected answer, whether it is a person, a location, or a time etc. To determine the answer type, the type of the question should be identified first. As mentioned earlier, knowing the question type addresses the "what to find" problem. Since most FBQA systems focus on *wh*- questions (*who, when, where, why, what*), it is natural to classify the types according to the stem of the question: the *wh*- words.

Most answers for *wh*- questions are related to named entities (NE); thus most FBQA systems classify the answers by different types of NE, such as: time, product, organization, person, etc. The NE identification technique from information extraction (IE) is quite helpful and usually is imported into this process. There are some other answer categories that do not belong to NE. As in Paşca's work [Paşca and Harabagiu, 2001b], type *reason* is applied to the *why* questions and type *definition* is included for questions asking for the definition of a concept.

A parser is often involved to find the answer type. For example, in Paşca's work, it depends on a concept hierarchy "Answer-Type Taxonomy" and a special-purpose parser.

- Answer-type taxonomy. The taxonomy is a tree structure constructed off-line which contains all the answer types that can be processed by the system. It is built in a top-down manner with general concepts on the top and more-specific concepts on lower levels. A subset of the taxonomy is shown in Figure 1.2.

  The top level of the hierarchy contains the most representative conceptual nodes, e.g. person, location, money, nationality, etc. Some of these are further categorized to more specific concepts. For instance, the location node is divided into university, city, country, etc. Some concepts are connected to corresponding synsets from WordNet. As an example, person is linked to several sub-trees rooted separately at scientist, performer, European, etc. It is worth noticing that although most concepts in the taxonomy are nouns, there are verbs and adjectives as well. A concept can

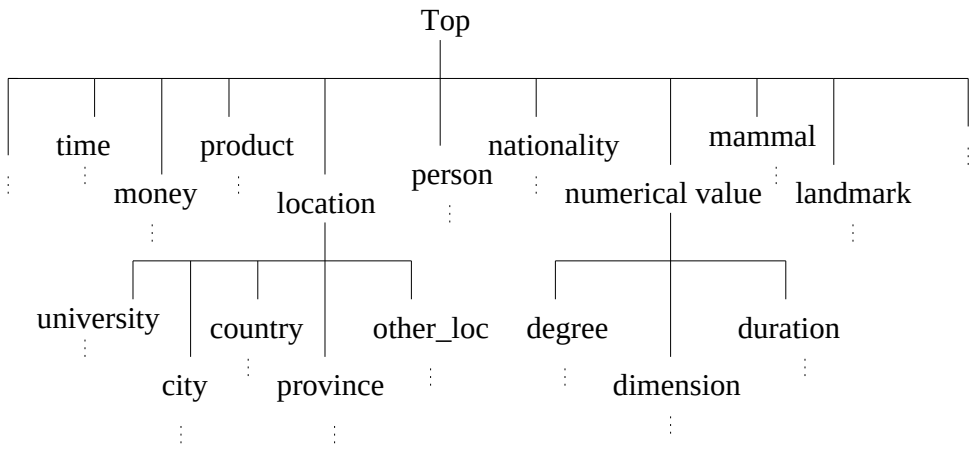


Figure 1.2: A subset of the answer-type taxonomy [Paşca and Harabagiu, 2001b]

be connected to the WordNet noun sub-hierarchies, verb sub-hierarchies or, adjectival satellites. For example, the product node is connected to nouns {*artifact, arteface*} and verbs {*manufacture, fabricate, construct*}. The whole taxonomy was constructed manually and was adapted to the sample questions.

- Parser. The *wh-* word in a question cannot always provide enough information on the type of the expected answer. For example, *what* can ask for many different types of things. To solve this problem, Paşca implemented a parser to find the word(s) in a question that help determine the expected answer type. The question is parsed to locate the word(s) that has a head-modifier dependency to the question stem (the *wh-* word). For instance, in the question *What do people usually buy in Hong Kong?*, *what* is a dependent of the verb *buy*. The word *buy* is then mapped onto the answer-type taxonomy to obtain the expected answer type, e.g. *product*.

This method is effective in determining the answer type at above 90% accuracy in the TREC test questions. However, there is a lot of manual work involved. Currently, the answer-type taxonomy encodes 8707 English concepts with 153 connections to WordNet sub-hierarchies [Paşca and Harabagiu, 2001b]. It will be quite burdensome to include more and more nodes into the taxonomy for the system to be adaptive to new expressions of questions and answers.

The approach explored by [Hovy et al., 2000] is similar to Paşca's work. In their "Webclopedia" system, they also built a taxonomy of answer types ("QA Typology") but WordNet

is not involved. The typology contains 94 nodes [Hovy et al., 2000]. An extended parser is also used in the process of answer-type identification, which contains some semantic background knowledge. A set of manually constructed rules is included in the parser to determine the correct answer type. The answer type produced by the parser can be the concepts in the QA Typology, P-O-S tags, roles produced in the parse tree, or concepts from the semantic type ontology of the parser.

Identification of question (answer) focus As defined by [Moldovan et al., 1999], "a focus is a word or a sequence of words which define the question and disambiguate it in the sense that it indicates what the question is looking for, or what the question is all about" (page 176). For example, the question *What type of bridge is the Golden Gate Bridge?* [Paşca and Harabagiu, 2001a] has *bridge* as the answer type and *type* as the answer focus. From the definition, the focus is very important for answering a question. Some systems mentioned the concept explicitly [Moldovan et al., 1999; Harabagiu et al., 2000; Ferret et al., 2001; Lee et al., 2001], others may include it in the general answer type identification process without discussing it separately. In both cases, no method or technique is provided to address the problem particularly.

Query generation The query-generation process usually involves keyword extraction from the original question with or without weights attached to them. For example, a query corresponding to the question *Who invented the paper clip?* is [paper AND clip AND invented]. Later, the query can be expanded by using a knowledge base such as WordNet. In some systems, after removing stop words, the keywords are selected by a set of heuristics [Moldovan and Harabagiu, 2000; Lee et al., 2001; Alpha et al., 2001]. Different systems have different preferences in whether lemmata or stemmed words should be used as keywords.

Query expansion is often applied to make sure that the correct answer will not be missed. Most systems use synonyms of the selected keywords in WordNet to expand the query. More sophisticated query-expansion techniques are explored in Harabagiu's work [Harabagiu et al., 2001a], which include three levels of alternations:

- Morphological alternations. When no answer is found by matching the original key-

words from the question, the morphological alternations are considered. For example, the noun *inventor* will be added to the query because of the original verb *invent*.

- Lexical alternations. WordNet is a source for adding lexical alternations to the query. In most cases, synonyms of a word are added, although other relationships may also be considered. For example, *killer* has a synonym *assassin* which should be included in the query expansion.

- Semantic alternations. The semantic alternations are defined as "the words or collocations from WordNet that (a) are not members of any WordNet synsets containing the original keyword; and (b) have a chain of WordNet relations or bigram relations that connect it to the original keyword" [Harabagiu et al., 2001a] (page 278). For example, the candidate words can be hypernyms or hyponyms of the original word, or even just related to it in some situation. To answer the question *How many dogs pull a sled in the Iditarod?*, since *sled* and *cart* are found to be forms of *vehicles*, the word *harness* that is related to *pull cart* is included in the query expansion.

Three heuristics are constructed to decide when and how to perform these alternations. However, for the semantic alternations, the heuristic does not specify which semantic relations should be considered in a particular situation (in fact, it is almost impossible to do so). This kind of problem seems to be an inherent limitation of knowledge-based approaches.

Matching of Question and Answer Keyword matching is the first criterion to filter out irrelevant answers in almost all FBQA systems. To make the system efficient, usually only text fragments that contain the query keywords will be returned instead of the whole documents. The number of returned fragments is fairly large, although it varies in different systems. The query keywords are expanded or shrunk to make sure that the proper number of fragments are returned.

In the next step, further matching is performed. Fragments that do not meet the strict requirements are filtered out. The filter can be used to verify the semantic relations or can be used as some ranking scheme. In Harabagiu's work, the filter is executed at three levels

[Harabagiu et al., 2001a]. At the first level, fragments that do not contain at least one concept of the same semantic category as the expected answer type are filtered out.

At the second level, the question represented in its semantic form produced by the parser in the question-processing phase is unified with the semantic forms of the fragments that contain the possible answers (answer candidates). The aim of the unification is to check how much information contained in the query is also contained in the answer candidates. Thus, the question concepts as well as the dependencies of the query terms which are represented by the semantic form are compared with the semantic forms of the answer candidates.

The semantic form of a sentence is derived from its syntactic parse tree. To construct the semantic form, the semantic concept that the sentence is about (the answer type) is added to the tree (which works as a slot in the question representation and the slot filler in the answer representation). Unimportant words are removed. Figure 1.3 is an example of the semantic forms:

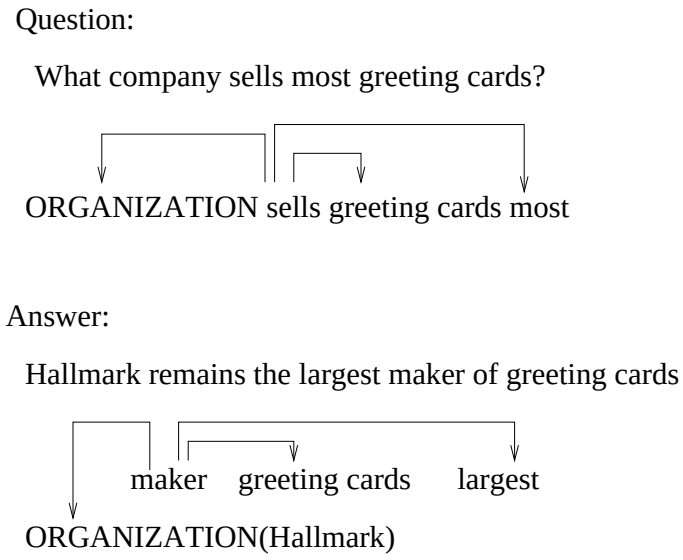


Figure 1.3: Semantic forms of sentences ([Paşca and Harabagiu, 2001b])

At the third level, the question and answer candidates are represented in their logical forms. The logical relations held by the terms in the query are evaluated in the abduction of answers. As an example of the logical form, the question *Why did Hong Li bring an umbrella?* can be represented as:

$$[REASON(x)\&Hong(y)\&Li(y)\&bring(e,x,y,z)\&umbrella(z)]$$

In this example, *Hong* and *Li* are identified as the same entity by using the same symbol *y*. A candidate is selected to be an answer if it can be proved by using the logical form. The prover processes the terms in the query logical form from the left to the right. For each term, it tries to identify corresponding information contained in the answer logical form. Real-world knowledge is needed here. For example, in the above question it may be helpful to know that *Hong Li* is a person's name.

In the work of [Hovy et al., 2001], answer type and answer focus are checked in the parse trees of the question and candidate answers in the matching process. When it is not enough for selecting a good answer, several heuristics are applied which consider the expected answer range, knowledge of abbreviation, and knowledge of some formats of special information (e.g. e-mail address, post code) etc.

Some systems try to choose the answer by ranking the candidate answers. The ranking often depends on heuristic rules about how the answer candidates contain the query terms, e.g., the order of the query terms in the answer candidates, the number of query terms that are matched, the distance from the position of the embedded answer type to the query terms etc. Some systems [Ferret et al., 2001; Prager et al., 2000; Srihari and Li, 1999; Alpha et al., 2001] implement the matching by ranking the passages according to weighted features or terms which are chosen off-line. Machine learning techniques are imported in the ranking in some systems. Paşca and Harabagiu [2001b] use a perceptron model to compare two candidates, while Prager et al. [2000] apply logistic regression to score NEs contained in the candidates.

Answer Extraction The task is to extract a concise answer from the answer candidates. In some systems, the candidates are strings in text windows of specified size [Moldovan et al., 1999]. Some others consider sentences as the candidates [Ferret et al., 2001; Hovy et al., 2001]. The candidates with the highest scores obtained in the matching process are extracted as the final answers. In knowledge-based systems, no particular techniques are applied to extract the answer.

Evaluation In the TREC-10 evaluation, the LCC system developed by Harabagiu et al. [2001b]performs well in both the main task and the list question task with the mean reciprocal rank (MRR)[1] of 0.57 in the main task and accuracy (no. of distinct instances/target no. of instances) of 0.76 in the list task. LCC is ranked second in the main task and first in the list task. The work of [Hovy et al., 2001] and [Alpha et al., 2001] are also in the top five ranked systems.

1.4.2 Data-Intensive Approaches

IE-based QA The NE identification techniques from IE are exploited in most FBQA systems. Some systems demonstrated that other techniques may also be helpful in QA. In the TREC-10 main task, the system that had the best performance was the one from InsightSoft-M [Soubbotin, 2001]. It applies a set of pre-defined patterns generated by analyzing the document collection to the answer candidates to match particular types of questions.

The idea is similar to pattern-matching and slot-filling in IE. Since various tasks should be addressed in a QA system, to take advantage of the pattern-matching technique, the Insight system classifies the questions into different categories and then constructs patterns for each category. There are two categories of patterns in the system:

- patterns representing a complete structure

  Example:
  capitalized word; parenthesis; four digits; dash; four digits; parenthesis
  would match “Mozart (1756 - 1791)”

- patterns composed by specific pattern elements

  Example:
  [number]+[term from currency list]
  would match “5 cents”

In the system, questions must be analyzed to obtain the accurate answer type so that the correct patterns will be triggered later. Relevant passages are obtained by searching for query

[1]MRR is an accuracy measure. To calculate the MRR, “an individual question receives a score equal to the reciprocal of the rank at which the first correct response is returned, or zero if none of the five responses contained a correct answer” [Voorhees, 2001].

keywords in the document collection. Only these passages are compared with the patterns to identify potential answers. As the patterns should adapt to various phrasing of answers, a great number of patterns are constructed manually for each question type (e.g., 23 patterns are built for the who-author type of questions). As in IE systems, manually constructing patterns is a very time-consuming task. So automatic pattern construction may be the future work of the pattern-matching-based systems.

As shown in the above examples, the system contains a set of pattern elements such as currency, person names, country names, etc. It seems that if an NE recognizer is used to replace these elements, the patterns can be simpler.

We can see that almost no deep knowledge analysis is involved in the pattern construction. However, proper knowledge is necessary in the system to process complicated questions. For example, the ambiguous question Who is Bill Gates? is actually asking for the reason why Bill Gates is famous. As in the knowledge-intensive approaches, the identification of the correct answer type is important in finding the correct answer here. It is declared by [Soubbotin, 2001] that detailed categorization of question types is a precondition for effective use of the method.

Since it is not possible to construct complete patterns for a type of question, for those cases in which the questions do not match any patterns, the system tries to select the answers by comparing lexical similarity of the question and answer candidates.

Another IE-based system was described in [Srihari and Li, 1999]. The idea is to answer questions by executing IE in three levels:

- Named entity: extract named entities as answer candidates
- Correlated entity: extract pre-defined relationships between the entities
- General events: extract different roles in general events (e.g. who did what to whom when and where)

However, only the first level was completed in the system.

From the above analysis we can see that IE-based systems lie somewhere between the knowledge-intensive and the data-intensive methods. The advantage of them is that the burden

of deriving and comparing the semantic and logic similarity of question and answer is released to some degree.

Redundancy-based QA The two other systems that also imported pattern-matching techniques into the query answer-matching process are MultiText [Clarke et al., 2000] and AskMSR [Dumais et al., 2002]. In MultiText, the patterns used consist of regular expressions with simple hand-coded extensions.

AskMSR just applies simple string-based manipulations in the query rewriting to formulate the patterns. The rewrite rule is a triple of the form [string, L/R/-, weight], where "string" is the reformulated search query, "L/R/-" indicates the position in the text where the answer is expected to find with respect to the query string (left, right or anywhere) and "weight" is a confidence figure for a particular query. If a query pattern is more likely to find the correct answer, it will have a higher weight than others. The following is an example [Dumais et al., 2002]:

Question: Who created the character of Scrooge?
Rewrite1: [*created + the character + of Scrooge*, left, 5]
Rewrite2: [*+the character + of Scrooge + was created + by*, right, 5]

However, the matching process is not the only component that helps find the answer in the two systems. The redundancy of the data is further explored to obtain the answer. The idea that data redundancy can be applied to question answering is basically the same in the two systems with slight differences. As indicated by [Clarke et al., 2001], the hypothesis was that correct answers could be distinguished from other candidates solely by their repeated occurrence in relevant passages.

The hypothesis is implemented by assigning weights to the candidate answers. In MultiText, after the pattern matching, the retrieved answer candidates are ranked according to the sum of the weights of the candidate answer terms that they contain. To calculate the term weights, [Clarke et al., 2001] used an idf-like formula with a redundancy parameter. The redundancy parameter is defined as the number of retrieved passages in which a particular term appears. In the answer generation process, the segment in a passage that maximizes the sum of the term weights it contains is extracted. MultiText ranked among the top five systems in the TREC-10 main task and list task evaluation.

In AskMSR, the n-grams (1-, 2-, 3- grams) in the retrieved passages are the candidates to be ranked. The weight of an n-gram depends on the confidence value of the rewrite rules that generated it ("5" in the above example of rewrite rules). The confidence values in all the unique retrieved passages in which the n-gram occurred are summed up to obtain the score of the n-gram. The n-grams are then filtered and re-weighted by a set of manually constructed heuristics. Finally, the remaining ones are tiled to get the answer. Tiling forms longer n-grams by merging overlapping shorter n-grams. For example, "A B C" and "B C D" is tiled into "A B C D." Compared with MultiText, the system does not need the corpus to be full-text indexed, nor does it need global term weights. However, AskMSR performs worse then MultiText. It was not in the top eight systems in the TREC-10 evaluation.

Although AskMSR was not so successful as MultiText in TREC-10, the idea that redundancy can help find the answer by using only simple patterns is verified by Dumais et al. [2002]. Their results show that the system performs much better on Web data than on the TREC data (the former is much larger than the latter). It is the same in MultiText. Compared with other approaches, a major contribution of redundancy-based methods is that they explore the relationships among good answer candidates. The correct answer of a question may be very difficult to identify because of its complicated formulation. However, it may be promoted by many relevant answer candidates that have simpler phrasings. This important information is ignored in other approaches.

Statistical QA Not many systems explored statistical approaches for FBQA. This might be because the potential of statistical models had not been realized in late 90's. Among the top-ranked systems in TREC-10, only one system [Ittycheriah et al., 2001] included a statistical model.

The system architecture in [Ittycheriah et al., 2001] is similar to the general architecture described in section 1.3. The statistical model is not applied to the whole system but rather on two components of the system: answer-type prediction and answer selection. The NE recognition is also implemented by using the statistical method. All three tasks are viewed as classification problems and the maximum-entropy models are constructed with three different feature sets. The answer types include the standard categories of NE in the Message Understanding Con-

ference (MUC) plus two more types: reason for why questions and phrase for all the others. The features take care of unigrams, bigrams, PoS, the position of the question words, as well as some expansion in WordNet. In answer selection, 31 features related to sentence, entity, definition, and linguistics (e.g. the answer candidate is either in the subject or object position etc.) are constructed. The NE annotation cares about the words, morphs, PoS, and grammar flags.

For the answer-type classification, 3300 questions were annotated manually before training. The training set for the answer-selection task is 400 question–answer pairs from TREC-8 and TREC-9. Because of the availability of the training data for the categories of NE, the answer types are almost confined to the MUC classes. The difficulty of obtaining enough training data is one problem that affects the system performance.

The statistical model works well in the answer-type identification task (accuracy 90.5%). The results for NE recognition are not reported in the paper, although it is indicated by error analysis that the performance is good. This system is one of the top-ranked systems in TREC-10, which indicates the effectiveness of using statistical models in FBQA.

1.4.3 Summary

Data-intensive approaches try to answer questions without deeply understanding the meaning of the questions and the answer text. This reduces the complexity of the system model. However, as we see from the above discussion, a pure data-based method is not enough to construct a system with high accuracy, because proper knowledge plays an important role in question analysis, which forms the guide in answer searching.

1.5 Non-factoid QA

In comparison to FBQA, NFQA is much less understood by researchers. However, it is such an important area that it is attracting more and more research interest [Niu et al., 2003; Diekema et al., 2003; Stoyanov et al., 2005; DUC, 2005].

NFQA deals with more complex information needs. We observe two distinct characteristics of NFQA as compared to FBQA.

- Non-factoid questions usually cannot be answered using a word or phrase, such as named entities. Instead, answers to these questions are much more complex, and often consist of multiple pieces of information from multiple sources.

- Compared to FBQA, in which an answer can be judged as *true* or *false*, NFQA needs to determine what information is *relevant* in answer construction.

Some examples of non-factoid questions are as follows.

In a patient with a generalized anxiety disorder, does cognitive behaviour or relaxation therapy decrease symptoms?

Was the most recent presidential election in Zimbabwe regarded as a fair election? [Stoyanov et al., 2005]

What advantages/disadvantages does an Aluminum alloy have over Ti alloy as the core for a honeycomb design? [Diekema et al., 2003]

Symptoms in the first question is a general concept, any clinical outcome of cognitive behaviour or relaxation therapy in anxiety disorder could be relevant. These outcomes could be different for different patient groups (e.g. different age groups); they may be positive in some clinical trials while negative in some others. All this evidence should be taken into account in constructing the answer. For the second question, it is not easy to reach an answer of *yes* or *no*. In fact, it might not be possible to do so, as it is very likely that both answers have supporters. Neither of them should be ignored in the answer. In addition, to either positive or negative attitude, information describing the reasons can be highly desirable. To answer the third question, we need to synthesize information on various aspects that the two metals are compared.

Because of the complex answers, current FBQA techniques will have difficulty in answering non-factual questions. Therefore, it is important to develop new strategies and techniques to address new challenges in NFQA.

1.5.1 Clinical question answering as NFQA

Clinicians often need to consult literature on the latest information in patient care, such as side effects of a medication, symptoms of a disease, or time constraints in the use of a medication.

The published medical literature is an important source to help clinicians make decisions in patient treatment [Sackett and Straus, 1998; Straus and Sackett, 1999]. Studies have shown that searching the literature can help clinicians answer questions regarding patient treatment [Gorman et al., 1994; Cimino, 1996; Mendonça et al., 2001]. It has also been found that if high-quality evidence is available in this way at the point of care—e.g., the patient's bedside—clinicians will use it in their decision making, and it frequently results in additional or changed decisions [Sackett and Straus, 1998; Straus and Sackett, 1999]. The practice of using the current best evidence to help clinicians in making decisions on the treatment of individual patients is called evidence-based medicine (EBM).

Questions posed by clinicians in patient treatment present interesting challenges to an NFQA system. For a clinical question, it is often the case that more than one clinical trial with different experimental settings will have been performed. Results of each trial provide some evidence on the problem. To answer such a question, all this evidence needs to be taken into account, as there may be duplicate evidence, partially agreed-on evidence, or even contradictions. A complete answer can be obtained only by synthesizing these multiple pieces of evidence, as shown in Figure 1.4. In our work, we take EBM as an example to investigate NFQA. Our targets are questions posed by physicians in patient treatment.

Clinical question: Are calcium channel blockers effective in reducing mortality in acute myocardial infarction patients?

Evidence1: . . . calcium channel blockers do not reduce mortality, . . . may increase mortality.

Evidence2: . . . verapamil versus placebo . . . had no significant effect on mortality.

Evidence3: . . . diltiazem significantly increased death or reinfarction.

Evidence4: . . . investigating the use of calcium channel blockers found a non-significant increase in mortality of about 4% and 6%.

Figure 1.4: Example of a clinical question, with corresponding evidence from *Clinical Evidence*.

1.5.2 Current research in NFQA

Unlike FBQA, in which the main research focuses on *wh-* questions (e.g. *when*, *where*, *who*) in a rather general domain, most work in NFQA starts with a specific domain, such as terrorism, or a specific type of question, such as opinion-related questions. The complexity of NFQA tasks may account for this difference. In this section, current work in NFQA is reviewed according to different research problems of the QA task that it addresses.

Question processing Because the information needs are more complex, some work put more efforts on understanding questions. Hickl et al. [2004], Small et al. [2004] and Diekema et al. [2003] suggest answering questions in an interactive way to clarify questions step by step. In addition, Hickl et al. argue that decomposition of complex scenarios into simple questions is necessary in an interactive system. As an example, the complex question *What is the current status of India's Prithvi ballistic missile project?* is decomposed into the following questions [Hickl et al., 2004]:

1. *How should 'India' be identified?*
2. *Pre-independence or post-independence, post-colonial, or post-1947 India?*
3. *What is 'Prithvi'?*
4. *What does Prithvi mean?*
5. *What class of missiles does Prithvi belong to?*
6. *What is its range/payload, and other technical details?*
7. ...

They propose two approaches to the decomposition: by approximating the domain-specific knowledge for a particular set of domains, and by identifying the decomposition strategies employed by human users. Preliminary results from two dialog pilot experiments suggest five strategies for question decomposition employed by experts that could be helpful in automatic decomposing complex questions.

Following that work, Harabagiu et al. [2004] derived intentional structure and the implicatures enabled by it for decomposing of complex questions, such as *What kind of assistance has North Korea received from the USSR/Russia for its missile program?* The authors claim that intentions that the user associate with the question may express a set of *intended questions*; and each intended question may be expressed as *implied questions*. The intended questions of this example include *What is the USSR/Russia? What is assistance? What are the missiles in the North Korean inventory?* Then, these intended questions further have implied questions, such as *Is this the Soviet/Russian government? Does it include private firms, state-owned firms, educational institutions, and individuals? Is it the training of personnel? What was the development timeline of the missiles?* Questions like *Will Prime Minister Mori survive the crisis?* and *Does Iraq have biological weapons?* are also questions that this paper is interested in [Harabagiu et al., 2004].

Two methods of generating the intentional structure of questions are explained by two examples in the paper. One is based on lexico-semantic knowledge bases (e.g. WordNet), and the other uses the predicate-argument structures of questions. The authors claim that the intentional structure may determine a different interpretation of the question, and answer extraction depends on the semantic relations between the coerced interpretations of predicates and arguments, although no details of evaluation are described in the paper.

The system HITIQA (High-Quality Interactive Question Answering) [Small et al., 2004] also emphasizes interaction with user to understand the information needs, although it does not attempt to decompose questions. During the interaction, the system asks questions to confirm the user's needs. After receiving *yes* or *no* from the user, the goal of searching is clearer. The interaction is data-driven in that questions asked by the system are motivated from previous results of information searching (which form the answer space).

Diekema et al. also suggest to have a question negotiation process for complex QA [Diekema et al., 2003]. The QA system deals with real-time questions related to "Reusable Launch Vehicles". For example, broad-coverage questions like *How does the shuttle fly?*, and questions about comparison of two elements such as *What advantages/disadvantages does an Aluminum alloy have over Ti alloy as the core for a honeycomb design?* are typical in the domain. A question-answering system architecture with a module of question negotiation between the

system and the questioner is proposed in the paper.

Matching of question and answer Berger et al. [2000] describe several interesting models to find the connection between question terms and answer terms.

- $tf \cdot idf$. This model is different from the standard $tf \cdot idf$ calculation. The conventional IR vector space model is applied in QA by taking the question and answer as different documents.

  Given an m-word question $q = \{q_1, q_2, ..., q_m\}$, and an n-word answer $a = \{a_1, a_2, ..., a_n\}$, the adapted cosine similarity between the question and the answer is given by the following formula [Berger et al., 2000]:

$$score(q, a) = \frac{\sum_{w \in q,a} \lambda_w^2 \cdot f_q(w) \cdot f_a(w)}{\sqrt{\sum_{w \in q} f_q(w)^2 \cdot \sum_{w \in a} f_a(w)^2}}, \tag{1.1}$$

  where

$$\lambda_w = idf(w) = \log\left(\frac{|D|}{|\{d \in D : f_d(w) > 0\}|}\right). \tag{1.2}$$

  where $f_d(w)$ is the number of times word w appears in document d. Here a document is an answer; D is the entire set of answers.

- Mutual information for query expansion. Instead of searching for terms for query expansion in a large knowledge taxonomy such as WordNet, a model for calculating the mutual information of query terms and answer terms is built. In this model, the mutual information of any pair of terms appearing in the training set of paired questions and answers is calculated. This can be used to locate the most relevant terms in the answer that are correlated with any question term. These terms are expected to be good candidates for expanding the query.

- Statistical translation model. Taking a machine translation view of the QA problem, the question and answer can be treated as two different languages. The model is built to learn how an answer a corresponds to a question q by calculating $p(q|a)$.

As indicated by [Berger et al., 2000], these models are presented for a problem slightly different from a typical QA task. It is to find answers within a large collection of candidate responses. The responses are supposed to be correct answers to the questions. It is not mentioned in the paper if there are only one-to-one relations between questions and answers. Since the answers are created according to the questions, it may be the case that the phrasing of question and answer has more overlap than it does in the general QA task. Also, answers to different questions may be easier to distinguish. Although such differences exist, the essence of the *answer-finding* task and the QA task are the same. Models explored by the former may adapt to the latter as well. Soricut and Brill [2006] extend Berger's work to answer FAQ-like questions. In their work, although FAQ question and answer pairs are used as training data, the goal is to extract answers from documents on the web, instead of pairing up existing questions and answers in FAQ corpora. Taking questions and answers as two different languages, a machine translation model is applied in the answer extraction module to extract three sentences that maximize the probability $p(q \mid a)$ (q is the question and a is the answer) from the retrieved documents as the answer.

In system HITIQA, frame structure is used to represent the text, where each frame has some attributes. For example, a general frame has *frame type*, *topic*, and *organization*. During the processing, frames will be instantiated by corresponding named entities in the text. In answer generation, text in the answer space is scored by comparing their frame structures with the corresponding goal structures generated by the system according to the question. Answers consist of text passages from which the zero conflict frames are derived. The correctness of the answers were not evaluated directly. Instead, the system was evaluated by how effective it is in helping users to achieve their information goal. The results of a three-day evaluation workshop validated the overall approach.

Cardie et al. [2003] aims to answer questions about opinions (multi-perspective QA), such as: *Was the most recent presidential election in Zimbabwe regarded as a fair election?*, *What was the world-wide reaction to the 2001 annual U.S. report on human rights?*. They developed an annotation scheme for low-level representation of opinions, and then proposed using opinion-oriented scenario templates to act as a summary representation of the opinions. Possible ways of using the representations in multi-perspective QA are discussed. In related work,

Stoyanov et al. [2005] analyzed characteristics of opinion questions and answers and show that traditional FBQA techniques are not enough for multi-perspective QA. Results of some initial experiments show that using filters that identify subjective sentences is helpful in multi-perspective QA.

Summary

The typical work discussed here shows the state-of-the-art in NFQA. Most systems are investigating complex questions in specific domains or of particular types. Although interesting views and approaches have been proposed, most work is at the initial stage, describing the general framework or potential useful approaches to address characteristics of NFQA.

As mentioned in section 1.1, our work on NFQA is in the medical domain. Clinical QA as an NFQA task, presents challenges similar to those of the tasks described in the previous subsection. Our work is to investigate these challenges by addressing a key issue: *what information is relevant*? We do not attempt to elicit such information by deriving additional questions, such as performing question decomposition [Hickl et al., 2004] or through interactive QA [Small et al., 2004]. Instead, we aim to identify the best information available in a designated source to construct the answer to a given question. The next chapter will describe our approach based on semantic class analysis.

1.6 Overview of contributions of thesis

This thesis focuses on a new branch of the question-answering task – NFQA. We show the difference between NFQA and FBQA by analyzing new characteristics of NFQA. We claim that answers to NFQA are usually more complex than named entities, and multiple pieces of information are often needed to construct a complete answer. We propose a novel approach to address these characteristics. Important subtasks in different modules of the new approach are identified, and automatic methods are developed to solve the problems.

To achieve these goals, we propose to use semantic class analysis in NFQA and use frame structure to represent semantic classes. We develop rule-based approaches to identify instances of semantic classes in text. Two important properties of semantic classes (*cores* and *polarity*)

are identified automatically. We show that the problem of relevance and redundancy in constructing answers is closely related to text summarization and build a summarization system to extract important sentences.

The QA approach based on semantic class analysis An event or scenario describes relations of several roles. Therefore, roles and their relations represent the gist of a scenario. We use semantic classes to refer to the essential roles in a scenario and propose an approach using semantic class analysis as the organizing principle to answer non-factoid questions. This approach contains four major components:

- Detecting semantic classes in questions and answer sources
- Identifying properties of semantic classes
- Question-answer matching: exploring properties of semantic classes to find relevant pieces of information
- Constructing answers by merging or synthesizing relevant information using relations between semantic classes

We investigate NFQA in the context of clinical question answering, and focus on three semantic classes that correspond to roles in the commonly accepted PICO format of describing clinical scenarios. The three classes are: the problem of the patient, the intervention used to treat the problem, and the clinical outcome. Interpretation of any treatment scenario can be derived using the three classes. This semantic class-based approach is described in Chapter 2.

Extracting semantic classes and analyzing their relations We use rule-based approaches to identify clinical outcomes and relations between instances of interventions in sentences. In QA, extracted clinical outcomes can be used directly to answer questions about outcomes of interventions. In the combination approach of outcome identification that we developed, a set of cue words that signal the occurrence of an outcome are collected and classified according to their PoS tags. For each PoS category, the syntactic components it suggests are summarized to derive rules for identifying boundaries of outcomes. This approach can potentially be applied to

identify or extract any semantic class. We identify six common relationships between different instances of interventions in a sentence and develop a cue-word based approach to identify the relations automatically. These relationships will improve accuracy of matching between questions and their answers. They can also improve document retrieval. After the index is built for these relations, they can be queried directly. Instances of semantic classes and their relations can be filled in predefined frame structures. Such information in free text is then represented by a more-structured data format that is easier for further processing. The combination approach and relation analysis are presented in Chapter 3.

Identifying cores of semantic classes We use the term *core* to refer to the smallest fragment of an instance of a semantic class that exhibits information rich enough for deriving a reasonably accurate understanding of the class. We found that cores are an important property of semantic classes as they can be the only clues to find the right answers. In Chapter 4, we show how cores of interventions, problems, and outcomes in a sentence can be identified automatically by developing an approach exploring semi-supervised learning techniques. This approach can be applied to identify cores of other semantic classes that have similar syntactic constituents, and it can be adapted to other semantic classes that have different syntactic constituents. This approach can potentially be applied to other classification problems that aim to group similar instances as well, e.g., word sense disambiguation. The concept of *cores* of semantic classes is pertinent to many tasks in computational linguistics. For example, *cores* are related to named entities. Some *cores* of semantic classes are named entities, while many are not. *Cores* as a new type of semantic unit extends the idea of named entities and the applications that rely on named entity identification.

Detecting polarity of clinical outcomes A clinical outcome may be positive, negative or neutral. Polarity is an inherent property of clinical outcomes. This information is mandatory to answer questions about benefits and harms of an intervention. Information on negative outcomes is often crucial in clinical decision making. We develop a method using a supervised learning model to automatically detect polarity of clinical outcomes. We show that this method has similar performance on different sources of medical text. We also identify a cause of the

bottleneck of performance using supervised learning approaches in polarity classification. The polarity detection task is discussed in Chapter 5.

Extracting components for answers We built explicit connection between text summarization and identifying answer components in NFQA, and construct a summarization system that explores a supervised classification model to extract important sentences for answer construction. We investigate the role of clinical outcomes and their polarity in this task. The system is presented in Chapter 6.

Chapter 2

Our approach for NFQA: semantic class analysis

As discussed in Chapter 1, answers in NFQA are not named entities and often consist of multiple pieces of information. In response to these major characteristics of NFQA, we propose to use frame-based semantic class analysis as the organizing principle to answer non-factual questions.

We investigated NFQA in the context of clinical question answering. In this chapter, we discuss the approach of semantic class analysis and how our work fits in the general QA framework.

2.1 Our approach of semantic class analysis

Clinical questions often describe scenarios. For example, they may describe relationships between clinical problems, treatments, and corresponding clinical outcomes, or they may be about symptoms, hypothesized disease and diagnosis processes. To answer these questions, essentially, we need an effective schema to understand scenario descriptions.

2.1.1 Representing scenarios using frames

Semantic roles Our principle in answering non-factual questions developed from the viewpoint that semantics of a scenario or an event is expressed by the semantic relationships between its participants, and such semantic relationships are defined by the role that each participant plays in the scenario. These relationships are referred to as *semantic roles* [Gildea and Jurafsky, 2002], or *conceptual roles* [Riloff, 1999]. This viewpoint can date back to frame semantics, posed by Fillmore [1976] as part of the nature of language. Frame semantics provides a schematic representation of events/scenarios that have various participants as roles. In our work, we use frames as our representation schema for the semantic roles involved in questions and answer sources.

Research on semantic roles has proposed different sets of roles ranging from the very general to the very specific. The most general role set consists of only two roles: PROTO-AGENT and PROTO-PATIENT [Dowty, 1991; Valin and Robert, 1993]. Roles can be more domain-specific, such as perpetrators, victims, and physical targets in a terrorism domain. In question-answering tasks, specific semantic roles can be more instructive in searching for relevant information, and thus more precise in pinpointing correct answers. Therefore, we take domain-specific roles as our targets.

The treatment frame Patient-specific questions in EBM usually can be described by the so-called *PICO format* [Sackett et al., 2000] in the medical domain. In a *treatment scenario, P* refers to the *status of the patient (or the problem), I* means an *intervention*, *C* is a *comparison intervention (if relevant)*, and *O* describes the *clinical outcome*. For example, in the following question:

> **Q:** In a patient with a suspected myocardial infarction does thrombolysis decrease the risk of death?

the description of the patient is *patient with a suspected myocardial infarction*, the intervention is *thrombolysis*, there is no comparison intervention in this question, and the clinical outcome is *decrease the risk of death*. Originally, *PICO format* was developed for therapy questions describing treatment scenarios and was later extended to other types of clinical questions such as diagnosis, prognosis, and etiology. Representing clinical questions with *PICO* format is

widely believed to be the key to efficiently finding high-quality evidence [Richardson et al., 1995; Ebell, 1999]. Empirical studies have shown that identifying *PICO* elements in clinical scenarios improves the conceptual clarity of clinical problems [Cheng, 2004].

We found that *PICO* format highlights several important semantic roles in clinical scenarios, and can be easily represented using the frame structure. Therefore, we constructed a frame based on it. Since *C* mainly indicates a comparison relation to *I*, we combined the comparisons as one filler of the same slot *intervention* in the frame, connected by a specific relation. We focus on therapy-related questions and built a *treatment frame* that contains three slots, as shown in Table 2.1.

Table 2.1: The treatment frame

P:	a description of the patient (or the problem)
I:	an intervention
O:	the clinical outcome

A slot in a frame designates a *semantic class* (corresponds to a *semantic role* or a *conceptual role*), and relations between semantic classes in a scenario are implied by the design of the frame structure. The treatment frame expresses a cause-effect relation: the *intervention* for the *problem* results in the *clinical outcome*.

When applying this frame to a sentence, we extract constituents in the sentence to fill in the slots in the frame. These constituents are *instances of semantic classes*. In this thesis, the terms *instances of semantic classes* and *slot fillers* are used interchangeably. Some examples of the instantiated treatment frame are as follows.

Sentence: One RCT [randomized clinical trial] found no evidence that low molecular weight heparin is superior to aspirin alone for the treatment of acute ischaemic stroke in people with atrial fibrillation.

P: acute ischaemic stroke in people with atrial fibrillation

I: low molecular weight heparin vs. aspirin

O: no evidence that low molecular weight heparin is superior to aspirin

Sentence: Subgroup analysis in people with congestive heart failure found that diltiazem significantly increased death or reinfarction.

P: people with congestive heart failure

I: diltiazem

O: significantly increased death or reinfarction

Sentence: Thrombolysis reduces the risk of dependency, but increases the risk of death.

P: —

I: thrombolysis

O: reduces the risk of dependency, but increases the risk of death

The first example states the result of a clinical trial, while the second and third depict clinical outcomes. We do not distinguish the two cases in this study, and treat them in the same manner.

How is it related to information extraction (IE)? Our approach of semantic class analysis has a close relation to IE, in which domain-specific semantic roles are often explored to identify predefined types of information from text [Riloff, 1999]. Our approach shares the same view with IE that semantic classes/roles are the keys to understand scenario descriptions. Frames are also used in IE as the representation scheme. Nevertheless, in our work, as shown by the above examples of treatment frames, the syntactic constituents of an instance of a semantic class can be much more complex than those of traditional IE tasks, in which slot fillers are usually named entities [Riloff, 1999; TREC, 2001]. Therefore, approaches based on such semantic classes go beyond named-entity identification, and thus will better adapt to NFQA. In addition, extracting instances of semantic classes from text is not the ultimate goal of QA. Frame representation of semantic classes provides a platform for matching between questions and answers in our QA system. We propose to conduct further analysis on semantic classes to search for answers to non-factual questions, which will be described in the following subsection.

2.1.2 Main components of a QA system guided by semantic class analysis

We propose to use semantic class analysis to guide the process of searching for answers to non-factual questions.

With semantic class analysis as the organizing principle, we identify four main components of our QA system:

- Detecting semantic classes in questions and answer sources
- Identifying properties of semantic classes
- Question-answer matching: exploring properties of semantic classes to find relevant pieces of information
- Constructing answers by merging or synthesizing relevant information using relations between semantic classes

To search for the answer to a question, the question and the text in which the answer may occur will be processed to detect the semantic classes. A semantic class can have various properties. These properties can be extremely valuable in finding answers, which we will discuss in detail in Chapter 4, 5, and 6. In the matching process, the question scenario will be compared to an answer candidate, and pieces of relevant information should be identified by exploring properties of the semantic classes. To construct the answer, relevant information that has been found in the matching process will be merged or synthesized to generate an accurate and concise answer. The process of synthesizing scenarios relies on comparing instances of semantic classes in these scenarios. For example, two instances are exactly the same or one is the hypernym of the other.

Scenario questions are common in other domains as well. For instance, questions about *shipping* events often depict relations between *provider*, *receiver*, and *means*; questions on events like *criticizing* often contain a *reviewer*, an *object*, the *reason*, and the *manner*. Frame semantics is a general representation schema for scenarios. Therefore, we expect that the main components in our QA approach can be applied to scenario questions in other domains rather easily.

2.1.3 The EPoCare Project

Our work is part of the EPoCare project ("Evidence at the Point of Care") at the University of Toronto. The project aims to provide clinicians fast access at the point of care to the best available medical information in published literature. Clinicians will be able to query sources that appraise the evidence about the treatment, diagnosis, prognosis, etiology, and prevalence of medical conditions. In order to make the system available at the point of care, the question-answering system will be accessible using hand-held computers. The project is an interdisciplinary collaboration that involves research in several disciplines. Project members in Industrial Engineering and Cognitive Psychology are investigating the design of the system through a user-centered design process, in which requirements are elicited from end users who are also involved in the evaluation of the prototypes. Project members in Knowledge Management and Natural Language Processing aim to ensure that the answers to queries are accurate and complete. And project members in Health Informatics will test the influence of the system on clinical decision-making and clinical outcomes.

Figure 2.1 shows the architecture of the system. There are three main components in the system. The **data sources** are stored in an XML document database. The **EPoCare server** uses this database to provide answers to queries posed by clinicians. The **knowledge base** is the source of medical terminologies.

Data sources The current data sources include the reviews of experimental results for clinical problems that are published in *Clinical Evidence* (CE) (version 7) [Barton, 2002], and *Evidence-based On Call* (EBOC) [Ball and Phillips, 2001].

- CE is a publication that reviews the current state of knowledge about the prevention and treatment of clinical conditions. It is a source of evidence on the effects of clinical interventions and it is updated every six months. The main content of CE is described in natural language. Evidence in CE is organized by a hierarchy structure of disease categories. In this structure, specific diseases are grouped together under each general category of disease, as shown in figure 2.2. For each specific disease, the effects of various interventions are summarized. CE is the text source that is used in most experiments reported in this thesis.

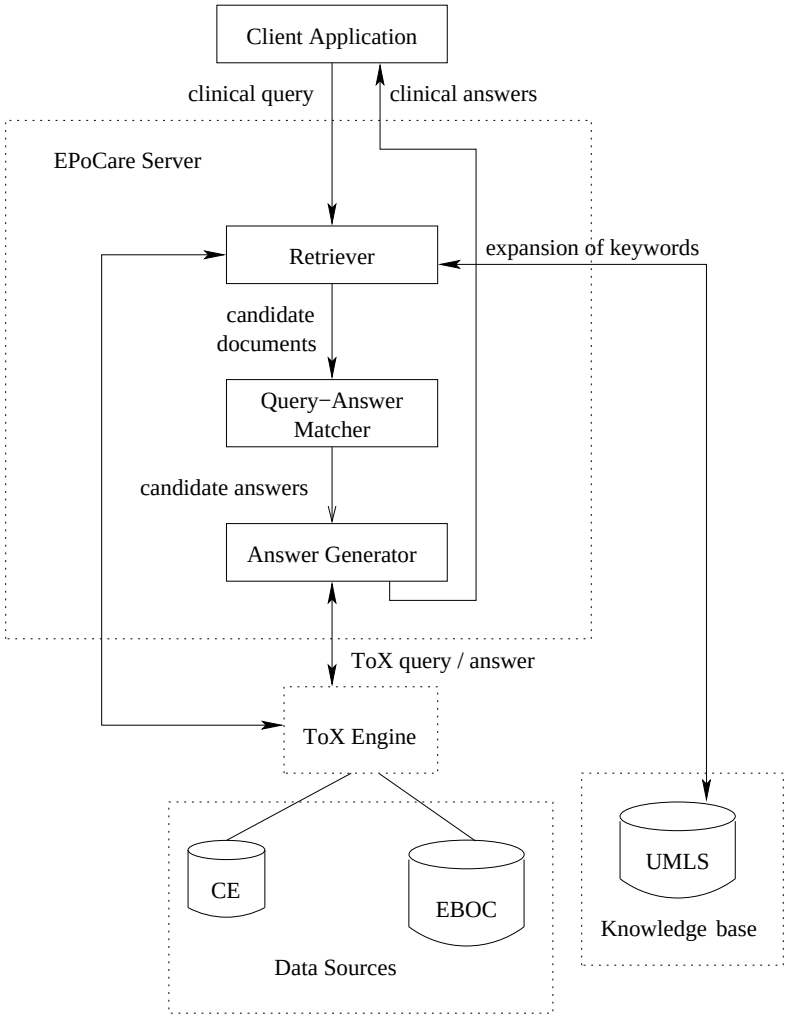


Figure 2.1: EPoCare system architecture.

- EBOC is another source that supports EBM. It provides the best available evidence on important topics in clinical practice by reviewing and summarizing knowledge in several databases, including the 'Best Evidence' CD-ROM, the Cochrane Library, and PubMed. Topics in EBOC are arranged alphabetically, indexed by disease area. Unlike CE, which has a focus on *treatments*, EBOC covers *prevalence*, *clinical features*, *investigations*, *therapy*, *prevention*, and *prognosis*. Summaries of the evidence are written in natural language, and are often accompanied by tables containing data derived from the original studies.

Both data sources are stored with XML mark-up in the database. The XML database is manipulated by ToX, a repository manager for XML data [Barbosa et al., 2001]. Repositories of distributed XML documents may be stored in a file system, a relational database, or remotely

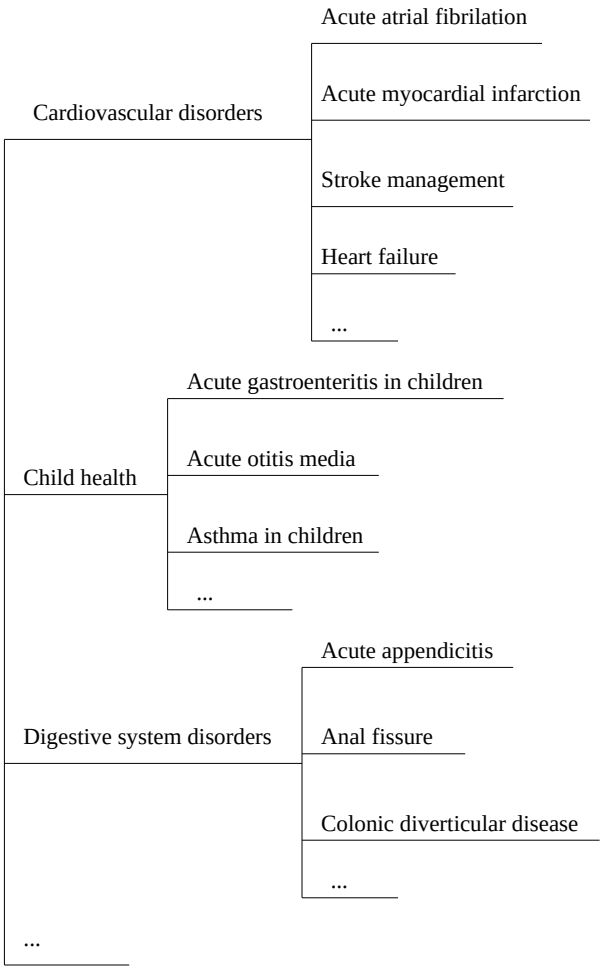


Figure 2.2: Disease categories in *Clinical Evidence*

on the Web. ToX supports document registration, collection management, storage and indexing choice, and queries on document content and structure.

EPoCare server In the EPoCare server, the Knowledge Management team takes care of keyword-based searching. A clinical query from the client is processed to form a database query of keywords. The query is sent by the retriever to the XML document database to retrieve relevant documents (e.g., a complete or partial section in CE) in the data sources using keyword matching. The results are then passed to the query–answer matcher to find the answer candidates. Finally, the best answer is determined and returned to the user.

The role of natural language processing is to allow the system to accept queries expressed in

natural language and to better identify answers in its natural-language data sources. After relevant documents are retrieved using the keyword-based matching, sentences in these documents will be processed using natural language processing techniques to find accurate and concise answers. Our work described in the following chapters can be adapted to several modules of the EPoCare system, including the *query-answer matcher* and the *answer extractor*.

Knowledge base The Unified Medical Language System (UMLS) is a knowledge base of medical terminologies. It is the major knowledge base in our work. UMLS contains three knowledge sources.

- The Metathesaurus is the central vocabulary component that contains information about biomedical and health-related concepts and the relationships among them. More than one name can be used to refer to the same concept. Metathesaurus links them together. There are 11 types of relationships between concepts in Metathesaurus, including *synonymy, broader,* and *narrower*. Each concept in the Metathesaurus is assigned to at least one semantic type from another component of UMLS – the Semantic Network.

- The Semantic Network is a network of the general categories or semantic types, such as *mental disability* and *pathological functions*, to which all concepts in the Metathesaurus have been assigned. It provides a consistent categorization of all concepts represented in the UMLS Metathesaurus and the important relationships between them. The 2003AA release of the Semantic Network contains 135 categories and 54 relations. In the Network, the categories are the nodes, and the relationships between them are the links. The primary link in the Network is the *isa* link. In addition, non-hierarchical relations are also identified, which belong to five major categories: *physically related to, spatially related to, temporally related to, functionally related to,* and *conceptually related to.*

- The SPECIALIST lexicon contains syntactic information about biomedical terms. It covers commonly occurring English words and biomedical vocabulary. The lexicon entry for each word or term records the syntactic, morphological, and orthographic information.

The following chapters discuss our work in three of the main components of our QA system. Figure 2.3 shows how this work fits in the general QA architecture.

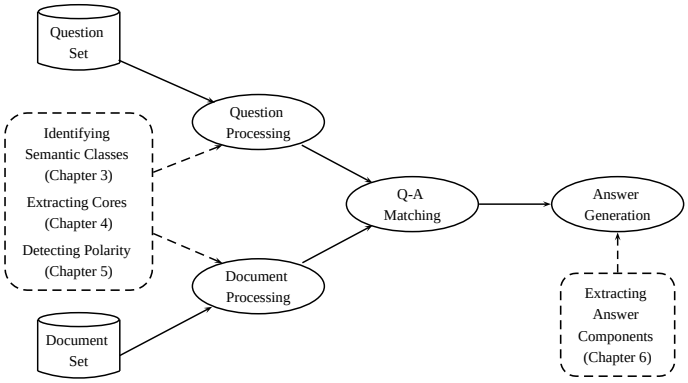


Figure 2.3: Our work in the QA framework

Chapter 3

Identifying semantic classes in text: filling the frame slots

This chapter discusses two problems in filling the treatment frame: identifying semantic classes in text and analyzing relations between instances of a semantic class. In semantic class identification, we focus on *clinical outcomes*, as outcomes are often expressed by more complex syntactic structures and are more difficult to label. In medical text, more than one intervention is often mentioned in the treatment of a disease, and various types of relations are involved between the interventions. These relations are analyzed automatically. We use rule-based approaches in these tasks.

3.1 Identifying clinical outcomes using a combination approach

In medical text, the appearance of some words is found often to be a signal of the occurrence of an outcome, and usually several words signal the occurrence of one single outcome. The combination approach that we applied for identifying outcomes is based on this observation. Our approach does not extract the whole outcome at once. Instead, it tries to identify the different parts of an outcome that may be scattered in the sentence, and then combines them to form the complete outcome.

In the combination approach, different pieces of an outcome are identified by some lexical identifiers, which are referred to as cue words. Each occurrence of a cue word suggests a portion of the expression of the outcome. Detecting all of them will increase the chance of obtaining the complete outcome. Also, different occurrences of cue words provide more evidence of the existence of an outcome. We evaluate the two phases of outcome identification separately. The first step is detecting the occurrence of outcomes, and the second is determining the boundaries of outcomes.

In the experiment, the text we use is from *Clinical Evidence* (CE). Two sections of CE were analyzed for detection of outcome. Outcome information in the text was annotated by a clinician. About two-thirds of each section (267 sentences in total) was taken as the analysis examples to construct the rules, and the rest (156 sentences) as the test set.

3.1.1 Detecting clinical outcomes in text

Collecting cue words We manually analyzed the analysis examples, and found that cue words of clinical outcomes belong to three PoS categories: noun, verb, and adjective. The cue words we found in the analysis are listed in Figure 3.1. All the inflectional variants of the cues are used as identifiers in the experiment.

Nouns: death benefit dependency outcome evidence harm difference risk deterioration mortality disability independence survival significance
Verbs: improve reduce prevent produce increase decrease affect
Adjectives: beneficial harmful negative adverse superior effective

---

Figure 3.1: Cue words for detecting clinical outcomes

In the following examples, cue words are highlighted.

(1) Thrombolysis *reduces* the risk of *dependency*, but *increases* the risk of *death*.

(2) Lubeluzole has also been noted to have *adverse outcome*, especially at higher doses.

Table 3.1: Results of identifying outcomes in CE

Method	Correct	False Positives	False Negatives	Precision%	Recall%	F-score%	Accuracy%
baseline	81	75	0	52 (81/156)	100	68	52
combination approach	67	14	14	83 (67/81)	83	83	82

(3) Several small comparative RCTs [randomized clinical trials] have found sodiumcromoglicate to be less *effective* than inhaled corticosteroids in *improving* symptoms and lung function.

(4) In the systematic review of calcium channel antagonists, indirect and limited comparisons of intravenous versus oral administration found no significant *difference* in *adverse* events.

The last two examples are different from other examples in that they express the outcomes of clinical trials, which we refer to as "results" in the following description when necessary. A "result" might contain a clinical outcome within it, as results often involve a comparison of the effects of two (or more) interventions on a disease.

Evaluating the outcome detection task We evaluated the cue word method of detecting the outcome on the test set. The result is shown in Table 3.1. A sentence that contains a clinical outcome is a positive case. Eighty-one sentences in the test set contain outcomes, which is 52% of all the test sentences. This was taken as the baseline of the evaluation: assigning all sentences in the test set to positive. By contrast, the accuracy of the cue word approach is 82%.

In error analysis, we found two main reasons that some outcomes were missed in the identification. One is that some outcomes do not have any cue word:

(5) *Gastrointestinal symptoms* and *headaches* have been reported with both montelukast and zafirlukast.

This example describes that two adverse events are associated with the treatment, which implies a negative clinical outcome. This outcome is only expressed implicitly, and therefore missed by the cue word-based approach.

The other reason is that although some words might be regarded as cue words, we did not include them in our set; for example, *fewer* and *higher*. Adjectives were found to have the most irregular usages in identifying outcomes. It is common for them to modify both interventions and outcomes, as shown in the following examples:

(6) Growth was significantly slower in children receiving *higher* dose inhaled corticosteroids.

(7) At 12 weeks, mean morning PEFR (peak expiratory flow rate) was 4% *higher* in the salmeterol group.

The word *higher* only signals a clinical outcome in the second example. Other adjectives such as *less, more, lower, shorter* and *longer* have similar problems. If they are taken as identifiers of outcomes then some false positives are very likely to be generated. However, if they are excluded, some true outcomes will be missed.

There were 14 false positives in the result of the experiment. The main cause was that some sentences contain cue words, and yet they did not provide any useful information:

(8) We found that the balance between *benefits* and *harms* has not been clearly established for the evacuation of supratentorial haematomas.

(9) The third systematic review did not evaluate these *adverse outcomes*.

As mentioned at the beginning of this subsection, currently, the cue words for detecting clinical outcomes were collected manually. In the next step of our work, we will investigate automatic approaches such as bootstrapping to find cues.

3.1.2 Determining the textual boundary of clinical outcomes

After the occurrence of clinical outcomes is detected using the cue words, the next problem is to determine their textual boundaries. Again, we rely on cue words to find the clue. As

mentioned before, a single clinical outcome often has several cue words in its expressions. The idea is that if the fragment of clinical outcome suggested by each cue word is identified, then the complete outcome can be obtained by combining or merging these fragments. Because the cue words belong to three PoS groups, we investigate the syntactic structure of the fragments in a clinical outcome that may be suggested by each of the groups.

Developing boundary detection rules In the analysis set, we obtained the syntactic structure of each sentence, as well as the PoS tags of words in the sentence using the Apple Pie parser [Sekine, 1997]. This parser produces phrase and PoS information that is needed for our work. It has various output formats, and one of them (shown in the following example) fits our task well and is easy to process. Some examples of the output of the parser with PoS and phrase information are listed in Figure 3.2. In the parsing trees, syntactic tags are all capitalized while

---

Sentence:

Thrombolysis reduces the risk of dependency, but increases the risk of death.

Output of the parser:

(S (NPL (NNPX Thrombolysis)) (VP (VP (VBZ reduces) (NP (NPL (DT the) (NN risk)) (PP (IN of) (NPL (NN dependency))))) (, -COMMA-) (CC but) (VP (VBZ increases) (NP (NPL (DT the) (NN risk)) (PP (IN of) (NPL (NN death)))))) (. -PERIOD-))

Sentence:

Lubeluzole has also been noted to have adverse outcome, especially at higher doses.

Output of the parser:

(S (NPL (NNPX Lubeluzole)) (VP (VBZ has) (ADVP (RB also)) (VP (VBN been) (VP (VBN noted) (TOINF (VP (TO to) (VP (VB have) (NPL (JJ adverse) (NN outcome)) (, -COMMA-) (PP (ADVP (RB especially)) (IN at) (NPL (JJR higher) (NNS doses)))))))) (. -PERIOD-))

Figure 3.2: Examples of output of Apple Pie Parser

words are not. Each word is attached with a PoS tag, which immediately precedes the word. Phrases are marked by higher level parentheses starting with phrase tags. Syntactic tags in the figure are explained in Appendix B.

Three general rules are derived to guide the boundary determination from the parsing trees of all sentences in the analysis example set:

- If a cue is a noun: The noun phrase that contains the noun will be part of the outcome.
- If a cue is a verb: The verb and its object (if the verb is in active voice) or its subject (if the verb is in passive voice) together constitute one portion of the outcome.
- If a cue is an adjective: The corresponding adjective phrase or the noun phrase belongs to the outcome.

In the first sentence in Figure 3.2, the verb *reduces* is a cue. According to the rule for verbs, the noun phrase that immediately follows it is part of the outcome. Therefore, *the risk of dependency* is included in the outcome. Similarly, the cue word *increases* identifies *the risk of death* as part of the outcome. By combining the two parts, the complete description of outcome is detected.

Cue words for the results of clinical trials are processed in a slightly different way. For example, for *difference* and *superior*, any immediately following prepositional phrase is also included in the results of the trial, as shown in Figure 3.3. The algorithm of the combination approach is described in Appendix D.

Sentence:

In the systematic review of calcium channel antagonists, indirect and limited comparisons of intravenous versus oral administration found no significant *difference* in adverse events.

Output of the parser:

(S (PP (IN In) (NP (NPL (DT the) (JJ systematic) (NN review)) (PP (IN of) (NPL (NN calcium) (NN channel) (NNS antagonists))))) (, -COMMA-) (NP (NPL (ADJP (JJ indirect) (CC and) (JJ limited)) (NNS comparisons)) (PP (IN of) (NPL (ADJP (JJ intravenous) (CC versus) (JJ oral)) (NN administration)))) (VP (VBD found) **(NPL (DT no) (JJ significant) (NN difference)) (PP (IN in) (NPL (JJ adverse) (NNS events))))** (. -PERIOD-))

Figure 3.3: An example of cue word *difference*

Table 3.2: Results of boundary detection of correctly identified outcomes in CE

Type of Overlap	Number	Percentage
Exact match	26	39
A entirely within B	19	28
B entirely within A	13	19
Both partially within the other	8	12
No match	1	1

A: Identified fragments; B: true boundary

Evaluating the boundary detection task Table 3.2 shows the result of boundary detection for those outcomes that were correctly identified (i.e., the true positives in the previous outcome detection task). The true boundary is the boundary of an outcome that was annotated manually. The *no match* case means that there is a true outcome in the sentence but the program missed the correct portions of text and marked some other portions as the outcome. The program identified 39% of the boundaries exactly the same as the true boundaries. In 19% of the samples, the true boundaries were entirely within the identified fragments. The spurious text in them (the text that was not in the true boundary) was found to be small in many cases, both in terms of number of words and in terms of the importance of the content. The average number of words correctly identified was 7 for each outcome and the number of spurious words was 3.4. The most frequent content in the spurious text was the intervention used to obtain the outcome. In the following examples, text in "⟨⟩" is the outcome (result) identified automatically, and text in "{}" is spurious.

(10) The RCTs found ⟨no significant adverse effects {associated with salmeterol}⟩.

(11) The second RCT also found ⟨no significant difference in mortality at 12 weeks {with lubeluzole versus placebo}⟩.

In the boundary detection task, again, adjectives are most problematic because of the great variation in the expression of outcomes they suggest. In the following examples, the true boundaries of outcomes are indicated by "[ ]", and adjectives are highlighted.

(12) Small RCTs with physiological rather than clinical end points found that giving $\beta 2$ agonists by metered dose inhaler with spacer to wheezy infants was [$\langle$ **effective** $\rangle$], [with less likelihood] than nebulisation [to show $\langle$ transient reduction of lung function $\rangle$].

(13) Nebulised $\beta 2$ agonists are known to cause [tachycardia, tremor, and hypokalaemia], but [$\langle$ serious **adverse** effects $\rangle$ are rare].

In sentence (12), the adjective *effective* is part of the outcome. In sentence (13), the clause that contains the adjective *adverse* is part of the outcome.

The correctness of the output of the parser also affects the performance, as shown in the following example:

Sentence:

RCTs found no evidence that lubeluzole improved clinical outcomes in people with acute ischaemic stroke.

Output of the parser:

(S (NPL (NNPX RCTs)) (VP (VBD found) (NPL (DT no) (NN evidence)) (NPL (DT that) (JJ lubeluzole) (JJ improved) (JJ clinical) (NNS outcomes)) (PP (IN in) (NP (NPL (NNS people)) (PP (IN with) (NPL (JJ acute) (JJ ischaemic) (NN stroke)))))) (. -PERIOD-))

The verb *improve* was incorrectly assigned to be an adjective in a noun phrase. Thus *improve* as a verb cue word was missed in identifying the outcome. However, another cue word *outcomes* was found, so the whole noun phrase containing *outcomes* was identified as the outcome. This example also shows that missing one cue word in identifying the outcome can be corrected by the occurrence of other cue words in the combination approach.

Related work

Machine-learning approaches and rule-based methods have been used for similar problems. Gildea and Jurafsky [2002] used a supervised learning method to learn both the identifier of the semantic roles defined in FrameNet such as theme, target, goal, and the boundaries of the roles [Baker et al., 2003]. A set of features were learned from a large training set, and

then applied to the unseen data to detect the roles. The performance of the system was quite good. However, it requires a large training set for related roles. It is usually expensive and time-consuming to obtain a large manually annotated data set.

Rule-based methods are explored in information extraction (IE) to identify roles to fill in slots in some pre-defined templates [Català et al., 2003]. The rules are represented by a set of patterns, and template role identification is usually conducted by pattern matching. Slots indicating roles are embedded in these patterns. Text that satisfies the constraints of a pattern will be identified, and the contents corresponding to the slots are extracted. This approach has been proved to be effective in many IE tasks. However, pattern construction is very time-consuming. In order to extract the roles and only the roles from text, their expressions have to be customized specifically in patterns. Targets consisting of complex syntactic constituents, e.g., clinical outcomes, will result in increasing difficulties in pattern construction, and less coverage of the patterns.

In our combination approach, instead of building one pattern to extract complete information of a target, as was done in most IE systems, we constructed simpler rules to identify portions of the target and then combine them to get the complete information. We expect this strategy to release some burden of manually creating patterns, especially for tasks having complex targets. In addition, since it is a rule-based approach, it does not need a large manually annotated training set. A limitation of this approach is that some connections between different portions of an outcome may be missing. Also, a different set of cue words may need to be collected when adapting to a new domain.

3.2 Analysis of Relations

More than one instance of semantic classes often occurs in a sentence; some instances are of the same class, while some of them are not. For those of different semantic classes, i.e., intervention, disease, and outcome, they often follow the relation implied by the frame structure. In our treatment frame, it is a cause-effect relation: the use of the intervention to the disease results in the outcome. For those of the same semantic class, e.g., intervention, we found that various relations occur. As discussed in Section 2.1, scenarios are about semantic classes and

their relations. In our approach of semantic class analysis, understanding such relations is an important part of interpreting scenarios. These relations are the target of this section. We only evaluate relations between different instances of intervention in a sentence, as we observe that a sentence often mentions more than one intervention. Relations between diseases can be analyzed in a similar way although they occur much less often than interventions.

Text from CE was analyzed manually to understand what relations are often involved and how they are represented. Then, an approach was developed to automatically identify the relations. The text for the analysis and test is the same as in the outcome identification task. Interventions in the text were annotated by a clinician.

Collecting cue words and symbols As with outcome identification, we found that these relations can be identified by a group of cue words or symbols. For example, the word *plus* refers to the COMBINATION of two or more interventions, the word *or*, as well as a comma, often suggests the ALTERNATIVE relation, and the word *versus* (or *v*) usually implies a COMPARISON relation, as shown in the following examples:

(14) The combination of aspirin *plus* streptokinase significantly increased mortality at 3 months.

(15) RCTs found no evidence that calcium channel antagonists, lubeluzole, aminobutyric acid agonists, glycine antagonists, *or* N-methyl-D-aspartate antagonists improve clinical outcomes in people with acute ischaemic stroke.

(16) One systematic review found no short or long term improvement in acute ischaemic stroke with immediate systemic anticoagulants (unfractionated heparin, low molecular weight heparin, heparinoids, *or* specific thrombin inhibitors) *versus* usual care without systemic anticoagulants.

It is worth noting that in CE, the experimental conditions are often explained in the description of the outcomes, for example:

(17) Growth was significantly slower in children receiving higher dose inhaled corticosteroids *(3.6cm, 95% CI 3.0 to 4.2 with double dose beclometasone v 5.1cm, 95% CI 4.5 to 5.7 with salmeterol v 4.5cm, 95% CI 3.8 to 5.2 with placebo).*

(18) It found that the addition for 4 weeks of oral theophylline versus placebo increased the mean number of symptom free days *(63% with theophylline v 42% with placebo; P=0.02)*.

(19) Studies of adults with poor control on low dose inhaled steroid *(see salmeterol v high dose inhaled corticosteroids under adult asthma)* have found greater benefit with additional long-acting $\beta 2$ agonists than with higher doses of inhaled steroid.

These conditions are usually in parentheses. They are often phrases and even just fragments of strings that are not represented in a uniform way with the other parts of the sentence. Their behavior is more difficult to capture and therefore the relations among the concepts in these descriptions are more difficult to identify. Because they usually are examples and data, omission of them will not affect the understanding of the whole sentence in most cases.

Six common relations and their cue words were found in the text which are shown in Table 3.3. Cue words and symbols between interventions were first collected from the training text. Then the relations they signal were analyzed. Some cue words are ambiguous, for example, *and*, and *with*. It is interesting to find that *and* in the text when it connects two interventions usually suggests an alternative relation rather than a combination relation, as in the example:

(20) Both salmeterol *and* beclometasone improved FEV1 compared with placebo, but the difference between beclometasone and salmeterol was not significant.

Compared with *versus, plus*, etc., *and* and *with* are weak cues as many of their appearances in the text do not suggest a relation between two interventions.

Experiment On the basis of this analysis, an automatic relation analysis process was applied to the test set. The test set is the same as in outcome identification. In the experiment, if a cue presents between two interventions in a sentence, the relation of the interventions will be detected. To deal with the case that more than one cue appear between two interventions, we assigned priorities to cue words/symbols according to how strong they are. A cue with higher priority determines the relation. *And* and *with* get lower priority compared to other cues. *And* has higher priority than *with*. For "," and "(", they are cues only when they are the only symbols

Table 3.3: Cue words/symbols for relations between interventions

Relation(s)	Cue Words/Symbols
COMPARISON	superior to, more than, versus, compare with, between ... and ...
ALTERNATIVE	or, ",", and
COMBINATION	plus, add to, addition of ... to ..., combined use of, and, with, "("
SPECIFICATION	with, "("
SUBSTITUTE	substitute, substituted for
PREFERENCE	rather than

between two interventions. Therefore they do not need to be assigned priorities. Other cues are not assigned priorities since they usually do not co-occur between two interventions. For ambiguous cues *and*, *with*, and *"("*, we assign the most frequent relation they indicate in the analysis examples to any occurrence of them in the test set. Therefore, *and* suggests alternative relations, *with* and *"("* indicate specification relations.

The test process was divided into two parts: one took parenthetical descriptions into account (case 1) and the other one did not (case 2). In the evaluation, for sentences that contain at least two interventions, "correct" means that the relation identified automatically is the same as marked by the annotator, "wrong" indicates that the two are different. In a "missing" case, a relation is ignored by the automatic approach. We did not evaluate the relation between any two interventions in a sentence; instead, we only considered two interventions that are related to each other by a cue word or symbol[1] (including those connected by cue words other than the set collected from the training text). The results of the two cases are shown in Table 3.4. Most errors are because of the weak indicators *with* and *and*. As in the outcome identification task, both the training and test sets are rather small, as no standard annotated text is available.

Some of the surface relationships in Table 3.3 reflect deeper relationships of the semantic

[1]There is only one implicit relation (a relation without a cue word identifier) for case 1 and case 2 respectively.

Table 3.4: Results of relation analysis

	Correct	Wrong	Missing	False Positive
case 1	49	7	10	9
case 2	48	7	3	6

classes. For example, COMPARISON, ALTERNATIVE, and PREFERENCE imply that the two (or more) interventions have some common effects on the disease(s) that are treated. The SPECIFICATION relation, on the other hand, suggests a hierarchical relation between the first intervention and the following ones, in which the first intervention is a higher-level concept and the following interventions are at a lower level. For example, in example (16), *systemic anticoagulants* is a higher-level concept, *unfractionated heparin*, *low molecular weight heparin*, etc., are examples of it that lie at a lower level.

3.3 Summary

This chapter describes our work in identifying clinical outcomes and analyzing relations between interventions. In question-answering, this information will be extracted to fill in the question frame and frames of potential answers. In addition, the relations can be indexed to improve document retrieval by supporting direct relation search. For example, if a user is interested in a comparison study of two interventions, then specifying both the relation and the interventions as the searching strategy will get more accurate results than just looking for the interventions. It can be very important for medical information retrieval, as such relations occur frequently in the text.

Chapter 4

Cores of semantic classes

In this chapter, we discuss a property of semantic classes – their cores.

In a frame structure, the slots in question and answer frames can be filled with either *complete* or *partial* information. Consider the following example, where parentheses delimit each instance of a semantic class (a slot filler) and the labels *P* (problem description), *I* (an intervention), *O* (the clinical outcome) indicate the type of the instance:

> *Sentence:*
>
> Two systematic reviews in (people with AMI)$_P$ investigating the use of (calcium channel blockers)$_I$ found a (non-significant increase in mortality of about 4% and 6%)$_O$.
>
> *Complete slot fillers*:
>
> P: people with AMI
>
> I: calcium channel blockers
>
> O: a non-significant increase in mortality of about 4% and 6%
>
> *Partial slot fillers*:
>
> P: AMI
>
> I: calcium channel blockers
>
> O: mortality

The partial slot fillers in this example contain the smallest fragments of the corresponding complete slot fillers that exhibit information rich enough for deriving a reasonably precise answer. We use the term *core* to refer to such a fraction of a slot filler (instance of a semantic

class).

4.1 Importance of cores

As discussed in Chapter 1, before the matching process, keyword-based document retrieval is usually performed to find relevant documents that may contain the answer to a given question. Keywords in the retrieval are derived from the question. Cores of semantic classes can be extremely valuable in searching for such documents for complex question scenarios, as shown in the following example.[1]

> *Question scenario:*
> A physician sees a 7-year-old child with asthma in her office. She is on flovent and ventolin currently and was recently discharged from hospital following her fourth admission for asthma exacerbation. During the most recent admission, the dose of flovent was increased. Her mother is concerned about the impact of the additional dose of steroids on her daughter's growth. This is the question to which the physician wants to find the answer.

For a complex scenario description like this, the answer could be missed or drowned in irrelevant documents found by inappropriate keywords derived from the question. However, the search can be much more effective if we have the information of cores of semantic classes, for example, *P: asthma, I: steroids, O: growth.*

Similarly, semantics presented in cores can help filter out irrelevant information that cannot be identified by searching methods based on simple string overlaps.

(21) In patients with **myocardial infarction**, do β **blockers** reduce all cause **mortality** and **recurrent myocardial infarction** without adverse effects?

(22) In someone with **hypertension** and **high cholesterol**, what management options will decrease his risk of **stroke** and **cardiac events**?

In question (21), the first occurrence of *myocardial infarction* is a disease and the second is part of the clinical outcome. In question (22), *stroke* is part of the clinical outcome rather than a disease to be treated, as it usually is. Obviously, string matching cannot distinguish between the

[1]The scenario is an example used in usability testing in the EPoCare project at the University of Toronto.

two cases. By identifying and classifying cores of semantic classes, the relations between these important semantic units in the scenarios are very clear. Therefore, documents or passages that do not contain *myocardial infarction* or *stroke* as clinical outcomes can be discarded.

In addition, identifying cores of semantic classes in documents can facilitate the question-answer matching process. Some evidence relevant to the above question scenario on *asthma* is listed below, where boldface indicates a core:

> Evidence1: A more recent systematic review (search date 1999) found three RCTs comparing the effects of **becolmetasone** and **non-steroidal medication** on linear **growth** in children with **asthma** (200 μg twice daily, duration up to maximum 54 weeks) suggesting a short-term decrease in linear **growth** of -1.54 cm a year.
>
> Evidence2: Two systematic reviews of studies with long term follow up and a subsequent long-term RCT have found no evidence of **growth retardation** in **asthmatic children** treated with inhaled **steroids**.

The evidence sentences here are from CE [Barton, 2002]. The clinical outcomes mentioned in the evidence have very different phrasings — yet both pieces of the evidence are relevant to the question. The pieces of evidence describe two distinct outcomes — that short-term decrease in growth is found and that there is no effect on growth in some long-term studies. Missing any of the outcomes will lead to an incomplete answer for the physician. Here, cores of the semantic classes provide the only clue that both pieces of this evidence are relevant to this question and should be included in the answer. Hence, a complete description of semantic classes does not have to be found. In fact, such a description with more information could make the matching harder to find because of the different expressions of the outcomes.

Finally, cores of semantic classes in a scenario are connected to each other by the relations embedded in the frame structure. The frame of the treatment scenario contains a cause-effect relation: an intervention used to treat a problem results in a clinical outcome.

In this chapter, we propose a method to automatically identify and classify the cores of semantic classes according to their context in a sentence. We take the treatment frame as an example, in which the goal is to identify cores of *interventions*, *problems*, and *clinical outcomes*. For ease of description, we will use the terms *intervention-core*, *disease-core*, and *outcome-core* to refer to the corresponding cores. We work at the sentence level, i.e., we

identify cores in a sentence rather than a clause or paragraph. Two principles are followed in developing the method. First, complete slot fillers do not have to be extracted before core identification. Second, we aim to reduce the need for expensive manual annotation of training data by using a semi-supervised approach.

4.2 Architecture of the method

In our approach, we first collect candidates of the target cores from sentences under consideration. For each candidate, we classify it as one of the four classes: *intervention-core*, *disease-core*, *outcome-core*, or *other*. In the classification, a candidate will get a class label according to its context, its UMLS semantic types, and the syntactic relations in which it participates. Figure 4.1 shows the architecture of the approach.

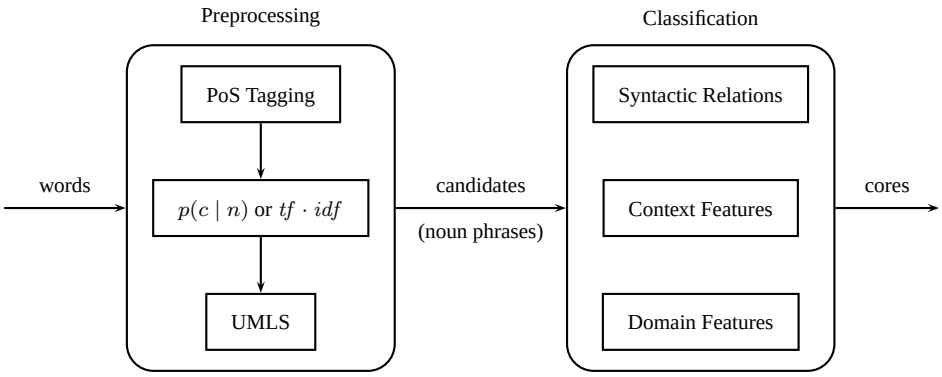


Figure 4.1: Architecture of the approach of core identification

4.3 Preprocessing

In the preprocessing, all words in the data set are examined. The first two steps are to reduce noise, in which some of the words that are unlikely to be part of real cores are filtered out. Then, the rest are mapped to their corresponding concepts, and these concepts are candidates of target cores.

PoS tagging Our observation is that cores of the three types of slot fillers are usually nouns or noun phrases. Therefore, words that are not nouns are first removed from the candidate set. PoS tags are obtained by using Brill's tagger [Brill, 1993].

Filtering out some *bad* nouns This step is the second attempt to remove noise from the candidate set. Nouns that are unlikely to be part of real cores are considered as *bad* candidates. Two research options of measures are used to evaluate how *good* a noun is.

- Extended $tf \cdot idf$.

  Let $DocSet = \{D_1, \cdots, D_d\}$ be a set of documents, and $NounSet = \{N_1, \cdots, N_n\}$ be the set of nouns in $DocSet$. For any $i \in [1..n]$ and $j \in [1..d]$, let $o_{ij}^{\#}$ be the number of occurrences of N_i in D_j, and $c_i^{\#}$ be the number of documents where N_i appears. The score of N_i is defined as

  $$score_i = max_{j \in [1..d]} \, (1 + tf_{ij}) \times idf_i$$

  where

  $$tf_{ij} = \begin{cases} \log o_{ij}^{\#} & : \quad o_{ij}^{\#} > 0 \\ 0 & : \quad o_{ij}^{\#} = 0 \end{cases} \quad \text{and} \quad idf_i = \log(d/c_i^{\#}).$$

  The formula $(1 + tf_{ij}) \times idf_i$ is taken from [Manning and Sch'utze, 1999], and it is the traditional measure of informativeness of a word with regard to a document. After this value is calculated for a noun in each document, the highest value of all the documents is taken as the final score of the noun. Nouns with scores lower than a threshold are removed from the candidate set. The threshold was set manually after observing the scores of some nouns that frequently occur in the text. *CE* text is used to get the score of a noun. For this, 47 sections in *CE* are segmented to 143 files of about the same size. Each file is treated as a document. This measure is referred to as $tf \cdot idf$ in later description.

- Domain specificity. We calculate the conditional probability $p(c \,|\, n) = p(c\,,\, n)/p(n)$, where c is the medical class, and n is a noun. It is the probability that a document is in the medical domain c given it contains the noun n. Intuitively, *intervention-cores*, *disease-cores*, and *outcome-cores* are domain-specific, i.e., a document that contains them is very likely to be in the medical domain. For example, *morbidity*, *mortality*, *aspirin*, and *myocardial infarction* are very likely to occur in a medicine-related context. This measure intends to keep highly medical domain-specific nouns in the candidate

set. A noun is a better candidate if the corresponding probability is high. Text from two domains is needed in this measure: medical text, and non-medical text. In our experiment, we use the same 47 sections in *CE* as the medical class text (separated into 143 files of about the same size). For the non-medical class, we use Reuters-21578 text collection. The collection has 21,578 documents. These documents appeared on the Reuters newswire in 1987, and they are collected 1000 to a file. We use it because newswire stories are mainly in the general domain. One file (1000 documents) in the Reuters collection are randomly selected for the calculation. Nouns whose probability values are below a threshold (determined in the same manner as in the $tf \cdot idf$ measure) are filtered out.

Mapping to concepts To this point, the candidate set consists of nouns. In many cases, nouns are part of noun phrases (concepts) that are better candidates of cores. For example, the phrase *myocardial infarction* is a better candidate of a disease-core than the noun *infarction*. Therefore, a noun is mapped to its corresponding UMLS concept in the sentence. All the concepts form the candidates of cores to be classified.

To find the concepts, a sentence is processed by the software MetaMap [Aronson, 2001]. MetaMap maps biomedical text to concepts in the UMLS Metathesaurus and finds their semantic types in the semantic network of UMLS. The major steps in the mapping conducted by MetaMap is outlined as follows.

1. Parsing. Text is parsed to get (mainly) simple noun phrases using the SPECIALIST minimal commitment parser [National Library of Medicine, 2004].

2. Variant generation. For each phrase, variants are generated using the knowledge in the SPECIALIST lexicon in UMLS and a database of synonyms. Various types of variants are generated for a phrase word, including its acronyms, abbreviations, synonyms, derivational variants, inflectional and spelling variants.

3. Candidate retrieval. Any string in Metathesaurus containing at least one of the variants is retrieved as a candidate concept.

4. Candidate evaluation. Each candidate is evaluated against the original phrase according to some weighting schema. The candidates are then ordered according to the mapping strength.

5. Mapping construction. Candidates involved in the phrase are combined to construct complete mappings, and the strength of the mappings is computed using the same schema in candidate evaluation. Mappings with the highest strength represent the best interpretation of MetaMap of the original phrase.

Figure 4.2 shows an example of the output of MetaMap. The sentence in the figure is processed by MetaMap and only part of the output is shown. The complete output is listed in Appendix E. In this example, MetaMap identified a concept *myocardial infarction* in the sentence, which is in correspondence to the candidate noun *infarction*. Therefore, the phrase *myocardial infarction* is used as a candidate, while the original noun *infarction* is not included in the new candidate set.

4.4 Representing candidates using features

We expect that candidates in the same semantic class will have similar behavior. Therefore, the idea of the classification is to group together similar candidates. The similarity is characterized by syntactic relations, context information, and semantic types in UMLS. All features are binary features, i.e., a feature takes value 1 if it is present; otherwise, it takes value 0.

Global syntactic relations Syntactic relations have been explored to group similar words [Lin, 1998] and words of the same sense in word sense disambiguation [Kohomban and Lee, 2005].

Given the following sentences [Lin, 1998]:

A bottle of *tezguino* is on the table.

Everyone likes *tezguino*.

Tezguino makes you drunk.

Sentence:

It found that the combined rate of myocardial infarction, stroke, or death was slightly lower in the lower dose than in the higher dose groups at 3 months.

Output of MetaMap:

Phrase: “of myocardial infarction”

Meta Candidates (6)

1000 Myocardial Infarction [Disease or Syndrome]

861 Infarction [Finding,Pathologic Function]

861 Myocardial [Functional Concept]

805 MI <2> (Without) [Qualitative Concept]

789 MIS (Mullerian duct inhibiting substance) [Amino Acid, Peptide, or Protein,Hormone]

789 Myocardium [Tissue]

Meta Mapping (1000)

1000 Myocardial Infarction [Disease or Syndrome]

Phrase: “stroke”

Meta Candidates (1)

1000 Stroke (Cerebrovascular accident) [Disease or Syndrome]

Meta Mapping (1000)

1000 Stroke (Cerebrovascular accident) [Disease or Syndrome]

Phrase: “or”

Meta Candidates (0): <none>

Meta Mappings: <none>

Figure 4.2: Example of output of MetaMap

We make *tezguino* out of corn.

Lin tried to infer that *tezguino* is similar to *beer*, *wine*, etc., i.e., it is a kind of drink, by comparing syntactic relations in which each word participates.

Kohomban and Lee [2005] determine the sense of a word in a context by observing a subset of all syntactic relations in the corpus that the word participates in. The hypothesis is that different instances of the same sense will have similar relations.

In our work, we need to group cores of the same semantic class. Such cores may participate in similar syntactic relations while those of different classes will have different relations. For example, intervention-cores often are subjects of sentences, while outcome-cores are often objects.

Candidates in our task are phrases, instead of words as in [Lin, 1998] and [Kohomban and Lee, 2005]. Thus, we extend their approaches of analyzing relations between two words to extract relations between a word and a phrase. This is done by considering all relations between a candidate noun phrase and other words in the sentence. To do that, we ignore relations between any two words in the phrase when extracting syntactic relations. Any relation between a word not in the phrase and a word in the phrase is extracted. We use the Minipar parser [Lin, 1994] to get the syntactic relations between words. After a sentence is parsed, we extract relevant syntactic relations from the output of the parser. A relation is represented using a triple that contains two words (one of them is in the noun phrase and the other is not) and the grammatical relation between them. Figure 4.3 shows relevant triples extracted from a sentence. The output of Minipar on this sentence is shown in Appendix F . Because long distance relations are considered, the relation between *thrombolysis* and *increases* is captured.

In the feature construction, a triple is taken as a feature. The set of all distinct triples is the syntactic relation feature set in the classification.

Local context Context of candidates is also important in distinguishing different classes. For example, a disease-core may often have *people with* in its left context. However, it is very unlikely that the phrase *people with mortality* will occur in the text.

As we mentioned, a sentence often contains several instances of semantic classes (hence several cores) that we are interested in. Wide-window context is not of much use in differenti-

Sentence:

Thrombolysis reduces the risk of dependency, but increases the chance of death.

Candidates:

thrombolysis, dependency, death

Relations:

(thrombolysis subj–of increase), (thrombolysis subj–of reduce)

(dependency pcomp-n–of of)

(death pcomp-n–of of)

Figure 4.3: Example of dependency triples extracted from output of Minipar parser.

ating these cores. In our experiment, we considered the two words on both sides of a candidate (stop words were excluded). When extracting context features, all punctuation marks were removed except the sentence boundary. The window did not cross boundaries of sentences.

We evaluated two representations of context: with and without order. In the ordered case, local context to the left of the phrase is marked by *-LLL*, that to the right is marked by *RRR-*. Symbols *-LLL* and *RRR-* are used only to indicate the order of text. For the candidate *dependency* in Figure 4.3, the context features with order are: *reduces-LLL*, *risk-LLL*, *RRR-increases*, and *RRR-chance*. The context features without order are: *reduces*, *risk*, *increases*, and *chance*.

This example shows a case where ordered context helps distinguish an intervention-core from an outcome-core. If order is not considered, candidates *thrombolysis* and *dependency* have overlapped context: *reduces* and *risk*. When taking order into account, they have no overlapped features at all – *thrombolysis* has features *RRR-reduces* and *RRR-risk*, while *dependency* has features *reduces-LLL* and *risk-LLL*.

Domain features As described in the *mapping to concepts* step in the preprocessing, at the same time of mapping text to concepts in UMLS, MetaMap also finds their semantic types. Each candidate has a semantic type defined in the Semantic Network of UMLS. For example, the semantic type of *death* is **organism function**, that of *disability* is **pathologic function**, and that of *dependency* is **physical disability**. These semantic types are used as features in the

Table 4.1: Number of Instances of Cores in the Whole Data Set

Intervention-core	Disease-core	Outcome-core	Total
501	153	384	1038

classification.

4.5 Data set

Two sections of *CE* were used in the experiments. A clinician labeled the text for intervention-cores and disease-cores. Complete clinical outcomes are also identified. Using the annotation as a basis, outcome-cores were labeled by the author. The number of instances of each class is shown in Table 4.1.

Data analysis In our approach, the design of the features is intended to group similar cores together. As a first step to verify how well the intention is captured by the features, we observe the geometric structure of the data.

In the analysis, candidates are derived using the domain specificity measure $p(c \mid n)$. Each candidate is represented by a vector of dimensionality D, where each dimension corresponds to a single feature. The feature set consists of syntactic features, ordered context, and semantic types. We map the high-dimensional data space to a low-dimensional space using the locally linear embedding (LLE) algorithm [Roweis and Saul, 2000] for easy observation. LLE maps high-dimensional data into a single global coordinate system of low dimensionality by reconstructing each data point from its neighbors. The contribution of the neighbors, summarized by the reconstruction weights, captures intrinsic geometric properties of the data. Because such properties are independent of linear transformations that are needed to map the original high-dimensional coordinates of each neighborhood to the low-dimensional coordinates, they are equally valid in the low-dimensional space. In Figure 4.4, the data is mapped to a 3-dimensional space (the coordinate axes in the figure do not have specific meanings as they do not represent coordinates of real data). Candidates of the four classes (intervention-core,

disease-core, outcome-core, and other) are represented by (red) stars, (blue) circles, (green) crosses, and (black) triangles, respectively. We can see that candidates in the same class are close to each other, and clusters of data points are observed in the figure.

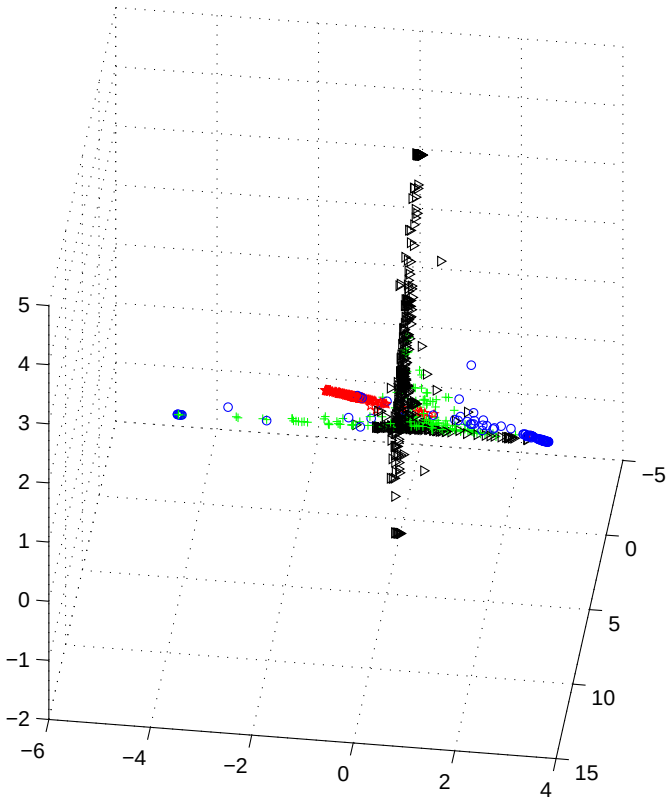


Figure 4.4: Manifold structure of data

4.6 The model of classification

Because our classification strategy is to group together similar cores and the cluster structure of the data is observed, we chose a semi-supervised learning model developed by Zhu et al. [2003] that explores the cluster structure of data in classification. The general hypothesis of this approach is that similar data points will have similar labels.

A graph is constructed in this model. In the graph, nodes correspond to both labeled and unlabeled data points (candidates of cores), and an edge between two nodes is weighted according to the similarity of the nodes. More formally, let $(x_1, y_1), \ldots, (x_l, y_l)$ be labeled data, where

$Y_L = y_1, \ldots, y_l$ are corresponding class labels. Similarly, let $(x_{l+1}, y_{l+1}), \ldots, (x_{l+u}, y_{l+u})$ be unlabeled data, where $Y_U = y_{l+1}, \ldots, y_{l+u}$ are labels to be predicted. A connected graph $G = (V, E)$ can be constructed, where the set of nodes V correspond to both labeled and unlabeled data points and E is the set of edges. The edge between two nodes i, j is weighted. Weights w_{ij} are assigned to agree with the hypothesis; for example, using a radial basis function (RBF) kernel: $w_{ij} = exp(-d^2(x_i, x_j)/\sigma^2)$, we can assign larger edge weights to closer points in Euclidean space.

Zhu et al. developed two approaches of propagating labels from labeled data points to unlabeled data points which have the same solution to the problem (the optimum solution is unique). One of them follows closely the intuition of the propagation, while the other is defined within a better framework. The first is described here to help understand the intuition of the model, and the second is depicted because it is used in the experiment.

The iteration approach In the prediction, labels are pushed from labeled points through edges to all unlabeled points using a probabilistic transition matrix, where larger edge weights allow labels to travel through easier. The $(l + u) \times (l + u)$ probabilistic transition matrix T is defined as [Zhu and Ghahramani, 2002]:

$$T_{ij} = \frac{w_{ij}}{\sum_{k=1}^{l+u} w_{kj}}$$

where T_{ij} is the probability moving from j to i. A label matrix B is a $(l+u) \times c$ matrix, where c is the number of classes in the task, and each row represents the label probability distribution of a data point.

In this problem setup, Zhu and Ghahramani proposed the label propagation algorithm:

1. Propagate $B \leftarrow TB$;
2. Row-normalize B to maintain the probability interpretation of the row;
3. Clamp the labeled data to keep the knowledge of originally labeled data;
4. Repeat from step 1 until B converges.

The label of a data point is determined by the largest probability in a row of B. It has been proved that the algorithm converges. In fact, the solution can be directly obtained without iterative propagation.

Label propagation using Gaussian random fields In [Zhu et al., 2003], Zhu et al. formulated the intuitive label propagation approach as a problem of energy minimization in the framework of Gaussian random fields, where the Gaussian field is over a continuous state space, instead of over discrete label set. The idea is to compute a *real-valued* function $f : V \rightarrow \mathcal{R}$ on graph G that minimizes the energy function $E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2$, where i and j correspond to data points in the problem. The function $f = argmin_f E(f)$ determines the labels of unlabeled data points. This solution can be efficiently computed by direct matrix calculation even for multi-label classification, in which solutions are generally computationally expensive in other frameworks.

This approach propagates labels from labeled data points to unlabeled data points according to the similarity on the edges, thus it follows closely the cluster structure of the data in prediction. We expect it to perform reasonably well on our data set. It is referred to as "SEMI" in the following description.

4.7 Results and analysis

We use SemiL [Huang et al., 2006], an implementation of the algorithm using Gaussian random fields in the experiment. SemiL provides different options for classification, among them some are pertinent to our problem setting:

- Distance type. The distance between two nodes can be either Euclidean distance or Cosine distance.

- Kernel type. The function used to assign weights on the edge. We use RBF in our experiment. The Sigma value in the RBF kernel is set heuristically using labeled data (Sigma is set to be the median of the distance from each data point in the positive class to its nearest neighbour in the negative class [Jaakkola et al., 1999].).[2]

- Normalization of the real-valued function f. It is designed to minimize the effect of unbalanced data set in the classification. As our data is unbalanced, we turn on this

[2]For heuristically set Sigma values in the thesis, several other Sigma values were used to verify the setting, and the results show that the performance is stable.

parameter to treat each class equally.

The performance of using Euclidean distance and Cosine distance in the similarity measure is compared in the experiment in Section 4.7.4. Default values are used for the rest of the parameters.

We first evaluate the performance of the semi-supervised model on different feature sets. Then, we compare the two candidate sets obtained by using $tf \cdot idf$ and domain specificity $p(c \mid n)$, respectively. Finally, we compare the semi-supervised model to a supervised approach to justify the usage of a semi-supervised approach in the problem.

In all experiments, the data set contains all candidates of cores. Unless otherwise mentioned, the result reported is achieved by using the candidate set derived by $p(c \mid n)$, the feature set of the combination of syntactic relations, ordered context, and semantic types, and the distance measure of cosine distance. The result of an experiment is the average of 20 runs. In each run, labeled data is randomly selected from the candidate set, and the rest is unlabeled data whose labels need to be predicted. We make sure all classes are present in labeled data. If any class is absent, we redo the sampling. The evaluation of the semantic classes is very strict: a candidate is given credit if it gets the same label as given by the annotator, and the tokens it contains are exactly the same as marked by the annotator. Candidates that contain only some of the tokens matching the labels given by the annotators are treated as the *other* class in the evaluation.

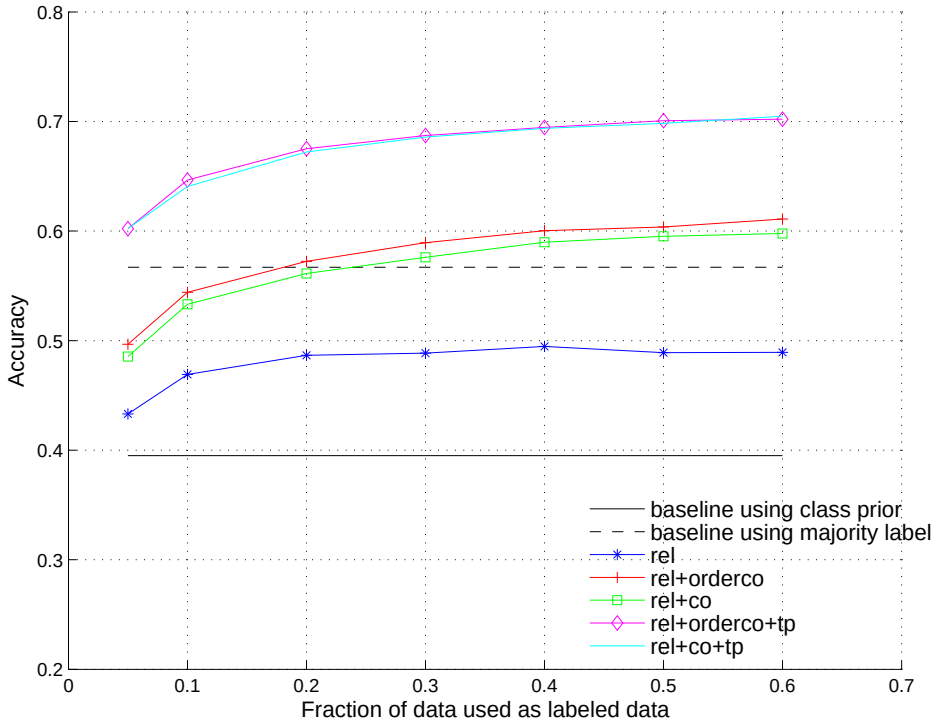
4.7.1 Experiment 1: Evaluation of feature sets

This experiment evaluates different feature sets in the classification. As described in section 4.3, two options are used in the second step of preprocessing to pick up *good* candidates. Here, as our focus is on the feature set we report only results on candidates selected by $p(c \mid n)$. The number of instances of each of the four target classes in the candidate set is shown in Table 4.2 (The performance of candidate selection will be discussed in subsection 4.7.2).

Figure 4.5 shows the accuracy of classification using different combinations of four feature sets: syntactic relations, ordered context, un-ordered context, and semantic types. We set a baseline by assigning labels to data points according to the prior knowledge of the distribution

Table 4.2: Number of Instances of Target Classes in the Candidate Set

Intervention-core	Disease-core	Outcome-core	Others	Total
298	106	209	801	1414



rel: syntactic relations

orderco: ordered context

co: no-order context

tp: semantic types

Figure 4.5: Classification Results of Candidates

of the four classes, which has accuracy of 0.395. Another choice of baseline is to assign the label of the majority class, *others* in this case, to each data point, which produces an accuracy of 0.567. However, all the three classes of interest have accuracy 0 according to this baseline. Thus, this baseline is not very informative in this experiment.

It is clear in the figure that incorporating new kinds of features into the classification results in a large improvement in accuracy. Only using syntactic relations (*rel* in the figure) as features,

the best accuracy is a little lower than 0.5, which is much higher than the baseline of 0.395. The addition of ordered context (*orderco*) or no-order context features (*co)*) improved the accuracy by about 0.1. Adding semantic type features (*tp*) further improved 0.1 in accuracy. Combining all three kinds of features achieves the best performance. With only 5% of data as labeled data, the whole feature set achieves an accuracy of 0.6, which is much higher than the baseline of 0.395. Semantic type seems to be a very powerful feature set as it substantially improves the performance on top of the combination of the other two kinds of features. Therefore, we took a closer look at the semantic type feature set by conducting the classification using only semantic types, and found that the result is even worse than using only syntactic relations. This observation reveals interesting relations of the feature sets. In the space defined by only one kind of features, data points may be close to each other, hence hard to distinguish. Adding another kind sets apart data points in different classes toward a more separable position in the new space. It shows that every kind of feature is informative to the task. The feature sets characterize the candidates from different angles that are complementary in the task.

We also see that there is almost no difference between ordered and unordered context in distinguishing the target classes, although ordered context seems to be slightly better when semantic types are not considered.

4.7.2 Experiment 2: Evaluation of candidate sets

In the second step of preprocessing, one of two research options can be used to filter out some *bad* nouns – using the $tf \cdot idf$ measure or the domain specificity measure $p(c \mid n)$. This experiment compares the two measures in the core identification task. A third option using neither of the two measures (i.e., skip the second step of preprocessing) is evaluated as the baseline. The first three rows in Table 4.3 are numbers of instances remaining in the candidate set after preprocessing. The last row shows the numbers of manually annotated true cores, which has been listed in Table 4.1 and is repeated here for comparison. We analyze the classification results using the candidate sets derived by $tf \cdot idf$, domain specificity, and baseline to evaluate the second step of preprocessing. Then, we compare the baseline to the manually annotated set of cores to evaluate the first and third steps of preprocessing.

Table 4.3: Number of Candidates in Different Candidate Sets

Measures	Intervention-core	Disease-core	Outcome-core	Others
$tf \cdot idf$	243	108	194	785
$p(c \mid n)$	298	106	209	801
baseline	303	108	236	1330
true cores	501	153	384	–

$tf \cdot idf$, domain specificity vs. baseline As shown in Table 4.3, there are much fewer instances in the *others* class in the sets derived by $tf \cdot idf$ and the probability measure as compared to those derived by the baseline, which shows that the two measures effectively removed some of the *bad* candidates of intervention-core, disease-core, and outcome-core. At the same time, a small number of real cores were removed. Compared to the baseline method, the probability measure kept almost the same number of intervention-cores and disease-cores in the candidate set, while omitting some outcome-cores. It indicates that outcome-cores are less domain-specific than intervention-cores and disease-cores. Compared to the $tf \cdot idf$ measure, more intervention-cores and outcome-cores were kept by the conditional probability measure, showing that the probability measuring the domain-specificity of a noun better characterizes the cores of the three semantic classes. The probability measure is also more robust than the $tf \cdot idf$ measure, as $tf \cdot idf$ relies more on the content of the text from which it is calculated. For example, if an intervention is mentioned in many documents of the document set, its $tf \cdot idf$ value can be very low although it is a good candidate of intervention-core.

The precision, recall, and F-score of the classification shown in Table 4.4 confirms the above analysis. The domain specificity measure gets substantially higher F-scores than the baseline for all the three classes that we are interested in, using different amounts of labeled data. Compared to $tf \cdot idf$, the performance of the domain specificity measure is much better on identifying intervention-core (note that $p(c \mid n)$ picked up more real intervention-cores than $tf \cdot idf$), and slightly better on identifying outcome-cores, while the two are similar on identifying disease-cores.

Table 4.4: Results of Classification on Different Candidate Sets

INT: *intervention-core* DIS: *disease-core* OUT: *outcome-core*

labeled data		1%			5%			10%			30%			60%		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
INT	baseline	.44	.69	.53	.51	.83	.63	.53	.87	.66	.58	.90	.70	.59	.92	.72
	$tf \cdot idf$	.44	.62	.51	.52	.74	.61	.55	.77	.64	.59	.84	.69	.60	.87	.71
	$p(c \mid n)$	.51	.65	**.57**	.60	.83	**.69**	.62	.86	**.72**	.65	.90	**.75**	.67	.91	**.77**
DIS	baseline	.16	.63	.25	.25	.68	.36	.31	.73	.43	.34	.84	.48	.35	.86	.49
	$tf \cdot idf$	.20	.55	**.29**	.31	.64	**.41**	.34	.70	.46	.39	.82	**.53**	.41	.86	**.55**
	$p(c \mid n)$	.18	.56	.27	.30	.66	**.41**	.34	.73	**.47**	.39	.83	**.53**	.41	.87	**.55**
OUT	baseline	.22	.42	.28	.33	.53	.41	.39	.61	.48	.44	.66	.53	.46	.69	.55
	$tf \cdot idf$	.30	.43	.35	.43	.56	**.49**	.47	.61	.53	.53	.66	.59	.55	.70	.61
	$p(c \mid n)$	.31	.46	**.37**	.43	.56	**.49**	.48	.62	**.54**	.54	.69	**.60**	.56	.71	**.63**

Baseline vs. the set of manually annotated cores As mentioned at the beginning of this subsection, the baseline candidate set was derived by the first (PoS tagging) and third step (mapping from nouns to concepts) in the preprocessing. As shown by Table 4.3, 62.3% of manually annotated cores are kept in the baseline. We roughly checked about one-third of the total true cores (manually annotated cores) in the data set and found that 80% of lost cores are because MetaMap either extracted more or less tokens than marked by the annotator, or it failed to find the concepts. 10% of missing cores are caused by errors of the PoS tagger, and the rest are because some cores are not nouns.

4.7.3 Experiment 3: Comparison of the semi-supervised model and SVMs

In the semi-supervised model, labels propagate along high-density data trails, and settle down at low-density gaps. If the data has this desired structure, unlabeled data can be used to help learning. In contrast, a supervised approach only makes use of labeled data. This experiment compares SEMI to a state-of-the-art supervised approach; the goal is to investigate how well unlabeled data contributes to the classification using the semi-supervised model. We compare

the performance of SEMI to support-vector machines (SVMs) when different amounts of data are used as labeled data.

Support vector machines In SVMs, the process of classification given a set of training examples is an optimization procedure searching for the optimal rule that predicts the label of unseen examples with minimum errors. The goal of classification is to infer a rule from a sample of labeled training examples so that it recognize new examples with high accuracy. More formally, the learner is given a training sample of n examples

$$(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n)$$

drawn according to an unknown but fixed distribution. Here $\vec{x}_i$ are patterns, y_i are labels or targets. In the classification, a function needs to be found based on training data:

$$f : \mathcal{X} \rightarrow \{^{+}_{-}1\},$$

where the domain $\mathcal{X}$ is some nonempty set that the patterns $\vec{x}_i$ are taken from, so that it will correctly classify unseen examples.

The goal of SVMs is to find an optimal hyperplane so that examples on the same side of the hyperplane will have the same label. SVMs learn decision functions:

$$f(\vec{x}) = sgn((\vec{w} \times \vec{x}) + b) = \begin{cases} +1 & : \quad \vec{w} \times \vec{x} + b > 0 \\ -1 & : \quad otherwise. \end{cases} \tag{4.1}$$

Each function corresponds to a hyperplane in the feature space. The classification task is then to determine on which side of the hyperplane a data point lies.

The optimal hyperplane that SVMs chose is the one with the largest *margin*. In the separable case, suppose we have a hyperplane that separates the positive examples from the negative examples. Let $d_+(d_-)$ be the shortest distance from the separating hyperplane to the closest positive (negative) example. The *margin* of such a hyperplane is $d_+ + d_-$. For the linear separable case, there is a $\vec{w'}$ and a b', such that all positive training examples lie on one side of the hyperplane, while all negative examples lie on the other side. In general, there can be more than one such hyperplanes, as shown in figure 4.6. Support vector machines choose the one with the largest margin (H' in the figure).

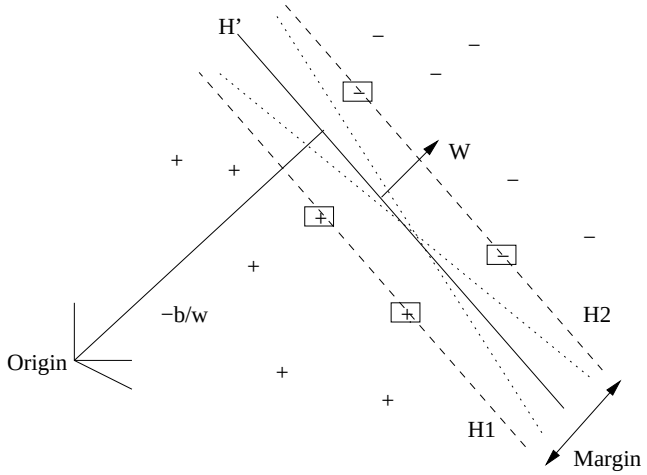


Figure 4.6: Linear separating hyperplanes in two dimensions. The support vectors are marked by squares.

In this experiment, we use OSU SVM [Ma et al., 2003], a toolbox for Matlab built on top of LIBSVM [Chang and Lin, 2001]. LIBSVM is an implementation of SVMs. We use RBF as the kernel method, and set the Sigma value heuristically using labeled data. SVM addresses the problem of unbalanced data using a parameter, which assigns weights to each class in the task. A class with larger weight will get more penalty when finding the optimum hyperplane. We set the parameter according to the prior knowledge of the class distribution and give larger weight to a class that contains less instances. Default values are used for other parameters.

Comparison of SEMI to SVMs As shown in Table 4.5, when there is only a small amount of labeled data (less than 5% of the whole data set), which is often the case in real-world applications, SEMI achieves much better performance than SVMs in identifying all the three classes. For intervention-core and outcome-core, with 5% data as labeled data, SEMI outperforms SVMs with 10% data as labeled data. With less than 60% data as labeled data, the performance of SEMI is either superior to or comparable to SVMs for intervention-core and outcome-core. It shows that SEMI effectively exploits unlabeled data by following the manifold structure of the data. The promising results achieved by SEMI show the potential of exploring unlabeled data in classification.

Table 4.5: F-score of Classification Using Different Models

Candidate set: produced by $p(c \mid n)$(see Table 4.2)

INT: *intervention-core* DIS: *disease-core* OUT: *outcome-core*

labeled data		1%	5%	10%	30%	60%
INT	SEMI	.57	.69	.72	.75	.77
	SVM	.33	.60	.68	.74	.77
DIS	SEMI	.27	.41	.47	.53	.55
	SVM	.21	.38	.54	.62	.65
OUT	SEMI	.37	.49	.54	.60	.63
	SVM	.07	.27	.44	.56	.62

Table 4.6: Accuracy using different distance measures.

Fraction of data as labeled data	10%	20%	30%	40%	50%	60%
Cosine distance	.647	.675	.687	.695	.701	.702
Euclidean distance	.341	.372	.405	.413	.410	.440

4.7.4 Experiment 4: Evaluation of distance measures

In the semi-supervised model, the cluster structure of the data is specified by the similarity of data points. Therefore, the choice of distance measure affects the performance of the classification. In this experiment, we compare two distance measures: cosine distance and Euclidean distance. Table 4.6 shows the classification accuracy using the two distance measures. The results show a large difference between them. Cosine distance is absolutely superior to Euclidean distance in the classification task.

The σ value in the RBF kernel is a scale parameter of the distance between two data points. Too-large a value can blur the distance between two well-separated points, while too-small a value may improperly enlarge the gap between data points. If σ is within a reasonable range, the performance of the classification will be relatively stable. Although parameter selection

was not a focus of the current work, we plot the results of using several different σ values in the classification in Appendix C to have some sense of the its effect.

4.8 Related work

The task of named entity (NE) identification, similar to the core-detection task, involves identifying words or word sequence in several classes, such as proper names (locations, persons, and organizations), monetary expressions, dates and times. NE identification has been an important research topic ever since it was defined in MUC [MUC, 1995]. In 2003, it was taken as the shared-task in CoNLL [Sang and Meulder, 2003]. Most statistical approaches use supervised methods to address the problem [Florian et al., 2003; Chieu and Ng, 2003; Klein et al., 2003]. Unsupervised approaches have also been tried in this task. Cucerzan and Yarowsky [1999] use a bootstrapping algorithm to learn contextual and morphological patterns iteratively. Collins and Singer [1999] tested the performance of several unsupervised algorithms on the problem: modified bootstrapping (DL-CoTrain) motivated by co-training [Blum and Mitchell, 1998], an extended boosting algorithm (CoBoost), and the Expectation Maximization (EM) algorithm. The results show that DL-CoTrain and CoBoost are superior to EM, while the two are almost the same.

Much effort in entity extraction in the biomedical domain has gene names as the target. Various supervised models including Naive Bayes, Support Vector Machines, Hidden Markov Models have been applied [Ananiadou and Tsujii, 2003]. The work most related to our core-identification in biomedical domain is that of Rosario and Hearst [2004], which extracts *treatment* and *disease* from MEDLINE and examines seven relation types between them using generative models and a neural network. They claim that these models may be useful when only partially labeled data is available, although only supervised learning is conducted in the paper. The best F-score of identifying *treatment* and *disease* obtained by using the supervised method is .71. Another piece of work extracting similar semantic classes is in [Ray and Craven, 2001]. They report an F-score of about .32 for extracting *proteins* and *locations*, and an F-score of about .50 for *gene* and *disorder*.

4.9 Summary

In this chapter, we identified an important property of semantic classes – the *core* and explained its role in matching a question to its answer. Then, we proposed a novel approach to automatically identify and classify cores of instances of semantic classes in scenario descriptions. A semi-supervised learning method was explored to reduce the need for manually annotated data.

In this approach, candidates of cores were first extracted from the text. We took two options to obtain a better candidate set by removing noise from the original set: $tf \cdot idf$ was used to find informative nouns, while a probability measure was to find domain-specific nouns. The results show that both measures effectively remove some noise, while the probability measure better captures the characteristics of cores. To do the classification, we designed several types of features and represented each candidate with the syntactic relations in which it participates, its context, and its semantic type, with the goal that candidates with similar representations are in the same class. Our experimental results show that syntactic relations work well together with other types of features. In the classification, a semi-supervised model that explores the manifold structure of the data was applied. The results show that the features characterize the cluster structure of the data, and unlabeled data is effectively used. We compared the semi-supervised approach to a state-of-the-art supervised approach, and showed that the performance of the semi-supervised approach is much better when there is only a small amount of labeled data, and performance of the two are comparable even when 60% of data are used as labeled data.

Our approach does not require prior knowledge of semantic classes, and it effectively exploits unlabeled data. The promising results achieved show the potential of semi-supervised models that explore the cluster structure of data in similar tasks. Features of syntactic relations and local context are general and can be used directly in tasks in other domain. The semantic type features make use of knowledge in UMLS, which is specific to medical domain. For tasks that have domain-specific knowledge bases like UMLS, similar features can be generated easily. For a domain without such knowledge base, the hierarchical information in WordNet can be used as a replacement, although it would be more difficult as the level of generalization needs to be determined.

A difficulty of using this approach, however, is in detecting boundaries of the targets. A

segmentation step that pre-processes the text is needed. This will be our future work, in which we aim to investigate approaches that perform the segmentation precisely.

As a final point, we want to emphasize the difference between cores and named entities. While the identification of NEs in a text is an important component of many tasks including question answering and information extraction, its benefits are constrained by its coverage. Typically, it is limited to a relatively small set of classes, such as *person*, *time*, and *location*. However, in sophisticated applications, such as the non-factoid medical question answering that we consider, NEs are only a small fraction of the important semantic units discussed in documents or asked about by users. As shown by the examples in this chapter, cores of clinical outcomes are often not NEs. In fact, many semantic roles in scenarios and events that occur in questions and documents do not contain NEs at all. For example, the *test method* in *diagnosis* scenarios, the *means* in a *shipping* event, and the *manner* in a *criticize* scenario may all have non-NE cores. Therefore, it is imperative to identify other kinds of semantic units besides NEs. Cores of semantic classes is one such extension that consist of a more diverse set of semantic units that goes beyond simple NEs.

Chapter 5

Polarity of Clinical Outcomes

One of the major concerns in patient treatment is the clinical outcomes of interventions in treating diseases: are they positive, negative or neutral? This polarity information is an inherent property of clinical outcomes. An example of each type of polarity taken from CE is shown below.

> *Positive*: Thrombolysis reduced the risk of death or dependency at the end of the studies.
>
> *Negative*: In the systematic review, thrombolysis increased fatal intracranial haemorrhage compared with placebo.
>
> *Neutral*: The first RCT found that diclofenac plus misoprostol versus placebo for 25 weeks produced no significant difference in cognitive function or global status.

Sentences that do not have information on clinical outcomes form another group: *no outcome*.

> *No outcome*: We found no RCTs comparing combined pharmacotherapy and psychotherapy with either treatment alone.

Polarity information is crucial to answer questions related to clinical outcomes. We have to know the polarity to answer questions about benefits and harms of an intervention. In addition, knowing whether a sentence contains a clinical outcome can help filter out irrelevant information in answer construction. Furthermore, information on negative outcomes can be crucial in clinical decision making.

In this chapter, we discuss the problem of automatically identifying outcome polarity in medical text [Niu et al., 2005]. More specifically, we focus on detecting the presence of a clinical outcome in medical text, and, when an outcome is found, determining whether it is positive, negative, or neutral[1]. We observe that a single sentence in medical text usually describes a complete clinical outcome. As a result, we perform sentence-level analysis in our work.

5.1 Related work

The problem of polarity analysis is also considered as a task of sentiment classification [Pang et al., 2002; Pang and Lee, 2004] or semantic orientation analysis [Turney, 2002]: determining whether an evaluative text, such as a movie review, expresses a "favorable" or "unfavorable" opinion. All these tasks are to obtain the orientation of the observed text on a discussion topic. They fall into three categories: detection of the polarity of words, sentences, and documents. Among them, as Yu and Hatzivassiloglou [2003] pointed out, the problem at the sentence level is the hardest one.

Turney [2002] has employed an unsupervised learning method to provide suggestions on documents as *thumbs up* or *thumbs down*. The polarity detection is done by averaging the semantic orientation (SO) of extracted phrases (phrases containing adjectives or adverbs) from a text. The document is tagged as *thumbs up* if the average of SO is positive, and otherwise is tagged as *thumbs down*. The SO is calculated by the difference in mutual information between an observed phrase and the positive word *excellent* and mutual information between the observed phrase and the negative word *poor*. Documents are classified as either positive or negative; no neutral position is allowed.

In more recent work, Whitelaw et al. [2005] explore *appraisal groups* to classify positive and negative documents. Similar to phrases used in Turney's work, *appraisal groups* consist of coherent words that together express the polarity of opinions, such as "extremely boring", or "not really very good". Instead of calculating the mutual information, a lexicon of *adjectival*

[1]This part of the work was carried out in collaboration with Xiaodan Zhu and Jane Li. They participated in the manual annotation. Xiaodan Zhu collected the BIGRAMS features, Jane Li collected the SEMANTIC TYPES.

appraisal groups (groups headed by an appraising adjective) is constructed semi-automatically. These groups are used as features in a supervised approach using SVMs to detect the sentiment of a document.

Pang et al. [2002] also deal with the task at document level. The sentiment classification problem were treated as a text classification issue and a variety of machine learning techniques were explored to classify movie reviews into positive and negative. Three classification strategies, Naive Bayes, maximum entropy classification, and support vector machines, were investigated, and a series of lexical features were employed on these classification strategies in order to find effective features. Pang et al. found that machine learning techniques can always outperform a human-generated baseline; among the three classification strategies, support vector machines perform the best and the Naive Bayes tends to be the worst; unigrams are the most effective lexical feature and indispensable compared with the other alternatives.

The main part of Yu and Hatzivassiloglou's work [Yu and Hatzivassiloglou, 2003] is at the sentence level, and is hence most closely related to our work. They first separate facts from opinions using a Bayesian classifier. Various features derived from observing semantic orientation of words are tried in this step. After opinion sentences are identified, they then use an unsupervised method to classify opinions into positive, negative, and neutral by evaluating the strength of the orientation of words contained in a sentence. A gold standard is built for evaluation, which includes 400 sentences labeled by one judge. On the task of distinguishing opinions from facts, the best performance is recall=0.92, precision=0.70 for the opinion class. The performance is much worse for the fact class. The best recall and precision obtained are 0.13 and 0.42. The unsupervised approach of detecting polarity of sentences achieves 0.62 accuracy.

The polarity information we are observing relates to clinical outcomes instead of the personal opinions studied by the work mentioned above. Therefore, we expect differences in the expressions and the structures of sentences in these two areas. For the task in the medical domain, it will be interesting to see if domain knowledge will help. These differences lead to new features in our approach.

5.2 A supervised approach for clinical outcome detection and polarity classification

As discussed in Section 5.1, various supervised models have been used in sentiment classification. At document-level, SVMs perform better than other models and achieve promising results [Pang et al., 2002]. In sentence-level analysis, Yu and Hatzivassiloglou [2003] use a Bayesian classifier to distinguish facts from opinions. The results for the fact class are not very satisfactory, which indicates that the task at sentence level may be more difficult. Since SVMs have been shown also very effective in many other classification tasks, in our work, we investigate SVMs in sentence-level analysis to detect the presence of a clinical outcome and determine its polarity.

In our approach, each sentence as a data point to be classified is represented by a vector of features. In the feature set, we use words themselves as they are very informative in related tasks such as sentiment classification and topic categorization. In addition, we use contextual information to capture changes described in clinical outcomes, and use generalized features that represent groups of concepts to build more regular patterns for classification.

We use binary features in most of the experiments except for the *frequency* feature in one of our experiments. When a feature is present in a sentence, it has a value of 1; otherwise, it has a value of 0. Among the features in our feature set, UNIGRAMS and BIGRAMS have been used in previous sentiment classification tasks, and the rest are new features that we developed.

5.2.1 Unigrams

A sentence is composed of words. Distinct words (unigrams) can be used as the features of a sentence. In previous work on sentiment classification [Pang et al., 2002; Yu and Hatzivassiloglou, 2003], unigrams are very effective. Following this work, we also take unigrams as features. We use unigrams occurring more than 3 times in the data set in the feature set, and they are called UNIGRAMS in the following description.

5.2.2 Context features

Our observation is that outcomes often express a change in a clinical value [Niu and Hirst, 2004]. In the following example, *mortality* was *reduced*.

(23) In these three postinfarction trials ACE inhibitor versus placebo significantly *reduced mortality, readmission for heart failure, and reinfarction.*

The polarity of an outcome is often determined by how a change happens: if a **bad** thing (e.g., mortality) was **reduced**, then it is a positive outcome; if a **bad** thing was **increased**, then the outcome is negative; if there is no change, then we get a neutral outcome. We tried to capture this observation by adding context features – BIGRAMS, two types of CHANGE PHRASES (MORE/LESS features and POLARITY-CHANGE features), and NEGATIONS.

BIGRAMS Bigrams (two adjacent words) are also used in sentiment classification. In that task, they are not so effective as UNIGRAMS. When combined with UNIGRAMS, they do not improve the classification accuracy [Pang et al., 2002; Yu and Hatzivassiloglou, 2003]. However, in our task, the context of a word in a sentence that describes the change in a clinical value is important in determining the polarity of a clinical outcome. Bigrams express the patterns of pairs, and we expect that they will capture some of the changes. Therefore, they are used in our feature set. As with UNIGRAMS, bigrams with frequency greater than 3 are extracted and referred to by BIGRAMS.

CHANGE PHRASES We developed two types of new features to capture the trend of changes in clinical values. The collective name CHANGE PHRASES is used to refer to these features.

To construct these features, we manually collected four groups of words by observing several sections in CE: those indicating **more** (*enhanced, higher, exceed, ...*), those indicating **less** (*reduce, decline, fall, ...*), those indicating **good** (*benefit, improvement, advantage, ...*), and those indicating **bad** (*suffer, adverse, hazards, ...*).

- MORE/LESS features. This type of feature emphasizes the effect of words expressing "changes". The way the features are generated is similar to the way that Pang et al. [2002] add negation features. We attached the tag _MORE to all words between the

more-words and the following punctuation mark, or between the **more**-words and another **more**(**less**) word, depending on which one comes first. The tag _LESS was added similarly. This way, the effect of the "change" words is propagated.

(24) The first systematic review found that β blockers significantly reduced_LESS the_LESS risk_LESS of_LESS death_LESS and_LESS hospital_LESS admissions _LESS.

(25) Another large rct (random clinical trial) found milrinone versus placebo increased _MORE mortality_MORE over_MORE 6_MORE months_MORE.

- POLARITY-CHANGE features. This type of feature addresses the co-occurrence of **more**/**less** words and **good**/**bad** words, i.e., it detects whether a sentence expresses the idea of "change of polarity". We used four features for this purpose: MORE GOOD, MORE BAD, LESS GOOD, and LESS BAD. As this type of features aims for the "changes" instead of "propagating the change effect", we used a smaller window size to build these features. To extract the first feature, a window of four words on each side of a **more**-word in a sentence was observed. If a **good**-word occurs in this window, then the feature MORE GOOD was activated (its value is set to 1). The other three features were activated in a similar way.

NEGATIONS Most frequently, negation expressions contain the word *no* or *not*. We observed several sections of CE and found that *not* often does not affect the polarity of a sentence, as shown in the following examples, so it is not included in the feature set.

(26) However, disagreement for uncommon but serious adverse safety outcomes has *not* been examined.

(27) The first RCT found fewer episodes of infection while taking antibiotics than while *not* taking antibiotics.

(28) The rates of adverse effects seemed higher with rivastigmine than with other anti-cholinesterase drugs, but direct comparisons have *not* been performed.

The case for *no* is different: it often suggests a neutral polarity or no clinical outcome at all:

(29) There are *no* short or long term clinical benefits from the administration of nebulised corticosteroids ...

(30) One systematic review in people with Alzheimer's disease found *no* significant benefit with lecithin versus placebo.

(31) We found *no* systematic review or RCTs of rivastigmine in people with vascular dementia.

We develop the NEGATION features to take into account the evidence of the word *no*. To extract the features, all the sentences in the data set are first parsed by the Apple Pie parser [Sekine, 1997] to get phrase information. Then, in a sentence containing the word *no*, the noun phrase containing *no* is extracted. Every word in this noun phrase except *no* itself is attached by a *_NO* tag.

5.2.3 Semantic types

Using category information to represent groups of medical concepts may relieve the data sparseness problem in the learning process. For example, we found that diseases are often mentioned in clinical outcomes as **bad** things:

(32) A combined end point of death or disabling stroke was significantly lower in the accelerated-t-PA group ...

Thus, all names of specific diseases in the text are replaced with the tag DISEASE.

Intuitively, the occurrences of semantic types, such as **pathologic function** and **organism function**, may be different in different polarity of outcomes, especially in the *no outcome* class as compared to the other three classes. To verify this intuition, we collect all the semantic types in the data set and use each of them as a feature. They are referred to as SEMANTIC TYPES. Thus, in addition to the words contained in a sentence, all the medical categories mentioned in a sentence are also considered.

The Unified Medical Language System (UMLS) is used as the domain knowledge base for extracting semantic types of concepts. The software MetaMap [Aronson, 2001] is incorporated for mapping concepts to their corresponding semantic types in the UMLS Metathesaurus.

5.3 Experiments

We carried out several experiments on two text sources: CE and Medline abstracts. Compared to CE text, Medline has a more diverse writing style as different abstracts have different authors. The performance of the supervised classification approach on the two sources is compared to find out if there is any difference. We believe that these experiments will lead to better understanding of the polarity detection task.

5.3.1 Outcome detection and polarity classification in CE text

Using CE as the text source, we evaluate a two-way classification task of distinguishing positive from negative outcomes, and the four-way classification of positive, negative, neutral outcomes, and no outcomes.

Positive vs. negative polarity

Experimental setup In this experiment, we have two target classes: positive outcomes and negative outcomes. The training and test sets were built by collecting sentences from different sections in CE; 772 sentences were used, 500 for training (300 positive, 200 negative), and 272 for testing (95 positive, 177 negative). All examples were labeled manually by the author.

We used the SVM^{light} implementation of SVMs [Joachims, 2002] to perform the classification and used the default values for the parameters.

Results and analysis Features used in the experiment are listed in the left-most column in Table 5.1. We construct features in two ways: using presence of a feature, a binary feature indicates whether a feature is present or not; and using frequency of a feature, the count of the number of occurrences of a feature in the sentence. The accuracies achieved by presence of

Table 5.1: Accuracy of positive/negative classification using a linear kernel in CE

Features	Presence (%)
baseline	65.1
UNIGRAMS	89.0
UNIGRAMS with DISEASE	90.1
UNIGRAMS with MORE/LESS	91.5
UNIGRAMS with DISEASE and MORE/LESS	92.7

features using a linear kernel (the default choice of kernels) are listed in Table 5.1. Frequency of features produces approximately the same results.

The baseline is to assign the negative label to all test samples as it is more frequent in the test set, which has the accuracy of 65.1%. As shown in the table, combining features achieves an accuracy as high as 92.7%. Using a more general category DISEASE instead of specific diseases has a positive effect on the classification. It is clear in the table that the MORE/LESS features improve the performance. Compared to using only UNIGRAMS, the combined feature set improves the accuracy by 0.037. The DISEASE and MORE/LESS features both contribute to distinguishing positive from negative classes.

A non-linear kernel RBF ($exp(-d^2(x_i, x_j)/\sigma^2)$) was also tested in SVMs. Using the feature set of presence of combining UNIGRAMS with DISEASE and MORE/LESS, the accuracy of the classification obtained with several σ values is shown in Appendix H. When σ is large, the performance is not very sensitive to its change, and becomes relatively stable.

Four-way classification

Experimental setup The data set of sentences in all the four classes was built by collecting sentences from different sections in CE (sentences were selected so that the data set is relatively balanced). The number of instances in each class is shown in Table 5.2. The data set is labeled manually by three graduate students, and each sentence is labeled by one of them. We used the OSU SVM package [Ma et al., 2003] with an RBF kernel for this experiment. The σ value

Table 5.2: Number of instances in each class (CE)

Positive	Negative	Neutral	No-outcome	Total
472	338	250	449	1509

was set heuristically using training data. Default values were used for other parameters in the package.

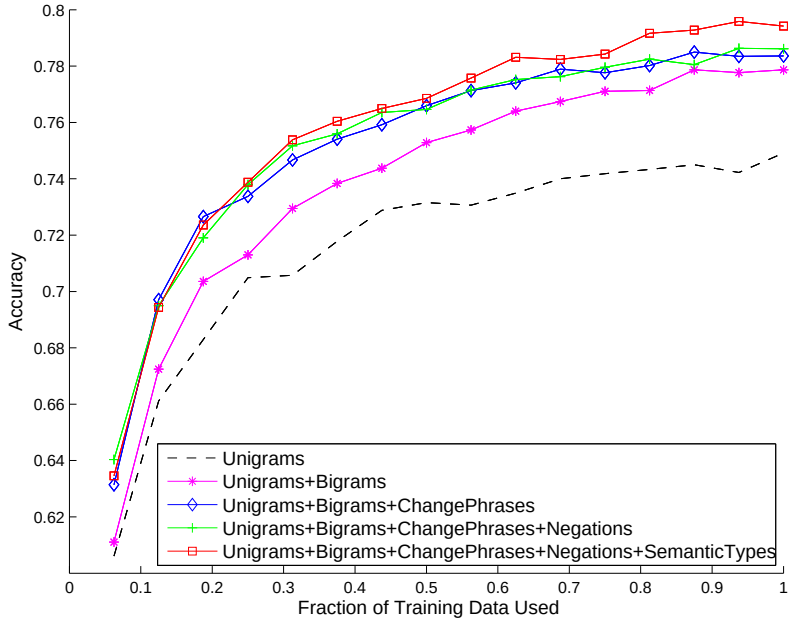


Figure 5.1: Accuracy of classification using different fractions of training data

Results and analysis

We first randomly select 20% of the whole data set as the test set (301 sentences), and use the rest (1208 sentences) as the training set. In the training process, we gradually add training samples until all of them are included, and observe the performance on the test set. The results are shown in Figure 5.1. As the figure indicates, accuracy goes up as more training data is used, and when more features are added. The complete feature set performs consistently the best. These results match our intuition that context information (BIGRAMS and CHANGE PHRASES) and

Table 5.3: Results of the four-way classification with different feature sets in CE

Features	Accuracy (%)	Relative Error Reduction (%) (to Unigrams)
(1)UNIGRAMS	76.9	–
(1)+(2)BIGRAMS	79.4	10.8
(1)+(2)+(3)CHANGE PHRASES	79.6	11.7
(1)+(2)+(3)+(4)NEGATIONS	79.6	11.7
(1)+(2)+(3)+(4)+(5)SEMANTIC TYPES	80.6	16.0

generalizations (SEMANTIC TYPES) are important factors in detecting the polarity of clinical outcomes.

Table 5.3 shows the results of the five feature sets used for classification. The accuracy is the average of 50 runs of the experiment. In each run, 20% of the data is selected randomly as the test set, and the rest is used as the training set. With just UNIGRAMS as features, we get 76.9% accuracy, which is taken as the baseline. The addition of BIGRAMS in the feature set results in an increase of about 2.5% in accuracy, which corresponds to 10.8% of relative error reduction. CHANGE PHRASES lead to a very small improvements and NEGATIONS do not improve the performance on top of BIGRAMS. This result seems to be different from the previous experiment of positive/negative classification, where the MORE/LESS features reduce the error rate. Note that CHANGE PHRASES intend to capture the impact of context, and bigrams also contain context information. It could be that some effect of CHANGE PHRASES has already been captured by bigrams. Also, since the target classes are different in the two tasks, CHANGE PHRASES may be more important in distinguishing positive from negative outcomes. The SEMANTIC TYPES features further improve the performance on top of the combination of other features, which shows that generalization is helpful.

Which class is the most difficult to detect, and why? To answer these questions, we further examine the errors in every class. The precision, recall and F-score of each class are shown in Table 5.4 (it is the result of one run of the experiment). It is clear in the table that the negative class has the lowest precision and recall. A lot of errors occur in distinguishing negative from

Table 5.4: Classification results of each class on CE data

	Positive	Negative	Neutral	No Outcome
Precision (%)	86.8	73.1	79.2	76.8
Recall (%)	83.2	73.1	76.0	82.0
F-score (%)	85.0	73.1	77.6	79.3

no-outcome classes. We studied the incorrectly classified sentences and found some interesting cases. Some of the errors are because descriptions of diseases in the no-outcome class are often identified as negative. These sentences are difficult in that they contain negative expressions (e.g., *increased risk*), yet do not belong to the negative class:

(33) Lewy body dementia is an insidious impairment of executive functions with Parkinsonism, visual hallucinations, and fluctuating cognitive abilities and increased risk of falls or autonomic failure.

Negative samples are sometimes assigned a positive label when a sentence has phrasings that seem to contrast, as shown in the following example:

(34) The mean increase in height in the budesonide group was 1.1 cm less than in the placebo group (22.7 vs 23.8 cm, P= 0005); . . .

In this sentence, the clinical outcome of impaired growth is expressed by comparing height increase in two groups, which is less explicit and hard to capture.

5.3.2 Outcome detection and polarity classification in Medline

With Medline abstracts, we evaluate two tasks: the first one is two-way classification that aims to detect the presence of clinical outcomes. In this task, a sentence is classified into two classes: containing a clinical outcome or not. The second task is the four-way classification, i.e., identifying whether an outcome is positive, negative, neutral, or a sentence does not contain an outcome.

Table 5.5: Number of instances in each class (Medline)

Positive	Negative	Neutral	No-outcome	Total
469	122	194	1513	2298

Table 5.6: Results of two-way and four-way classification with different feature sets (Medline)

RER=Relative Error Reduction (compared to unigrams)

Features	two-way		four-way	
	Accuracy (%)	RER (%)	Accuracy (%)	RER (%)
(1) UNIGRAMS	80.1	–	75.5	–
(1)+(2) BIGRAMS	81.7	8.0	77.4	7.8
(1)+(2)+(3) CHANGE PHRASES	82.0	9.5	77.6	8.6
(1)+(2)+(3)+(4) NEGATIONS	81.9	9.0	77.6	8.6
(1)+(2)+(3)+(4)+(5) SEMANTIC TYPES	82.5	12.1	78.3	11.4

Experimental setup We collected 197 abstracts from Medline that were cited in CE. The number of sentences in each class is listed in Table 5.5. The data set was annotated with the four classes of polarity information by two graduate students. Each single sentence is annotated by one of them. In this experiment, again, 20% of the data was randomly selected as test set and the rest was used as the training data. The averaged accuracy was obtained from 50 runs. We used the same SVM package as in Section 5.3.1 for this experiment, parameters were set in the same manner.

Results and analysis Results of the two tasks are shown in Table 5.6. Not surprisingly, the performance on the two-way classification is better than on the four-way task. For both tasks, we see a similar trend in accuracy as in CE text (see Table 5.3). The accuracy goes up as more features are added, and the complete feature set has the best performance. Compared to UNIGRAMS, the combination of all features significantly improves the performance in both tasks (paired t-test, p values < 0.0001). With just UNIGRAMS as features, we get 80.1% accu-

racy for the two-way task. The addition of BIGRAMS in the feature set results in a decrease of 1.6% in the error rate, which corresponds to 8.0% of relative error reduction as compared to UNIGRAMS. Similar improvements are observed in the four-way task. The SEMANTIC TYPES features also slightly reduce the error rate.

Compared to the results on CE text in Table 5.3, the four-way classification task tends to be more difficult on Medline text. This can be observed by comparing the improvement of adding all other features to UNIGRAMS. As we mentioned in section 5.3, Medline abstracts have a more diverse writing style because they are written by different authors. This could be a factor that makes the classification task more difficult. However, the general performance of features on Medline abstracts and CE text is similar, which shows that the feature set is relatively robust.

In our outcome detection and polarity classification task, UNIGRAMS are very effective features, as has been previously shown in the context of sentiment classification problems. This shows that information in words is very important for the polarity detection task. Context information represented by BIGRAMS and CHANGE PHRASES is also valuable in our task (see Table 5.1, Table 5.3, and Table 5.6). The effectiveness of BIGRAMS is different from the results obtained by Pang et al. [2002] and Yu and Hatzivassiloglou [2003]. In their work, adding bigrams does not make any difference in the accuracy, or even is slightly harmful in some cases. This indicates the difference in the expression of polarity in clinical outcomes and the polarity in opinions. Generalization features (DISEASE in Table 5.1, SEMANTIC TYPES in Table 5.3 and Table 5.6) are also helpful in our task.

5.4 Discussion

The performance bottleneck in polarity classification As described in Section 5.1, supervised approaches have been used in sentiment classification. Features used in these approaches usually include: n-grams, PoS tags, and features based on words with semantic orientations (e.g., adjectives such as *good, bad*). In all such studies, a common observation is that unigrams are very effective, while adding more features does not gain much.

- In the task of detecting polarity of documents [Pang et al., 2002], the best performance is obtained using unigrams.

- In the sentence-level opinion/fact classification task [Yu and Hatzivassiloglou, 2003], as described in section 5.1, various features based on semantic orientation of words are tried, including counts of semantically oriented words, the polarity of the head verbs and the average semantic orientation score of the words in the sentence. A gold standard set is built which includes 400 sentences labeled by one judge. In the opinion class, the only result better than the performance of unigrams is obtained by combining all features, which results in only 0.01 improvement in precision. Similarly, not much is achieved by adding all other features in detecting facts.

- In [Whitelaw et al., 2005], the best performance of the approach is achieved by the combination of unigrams with the appraisal groups, which is 3% higher in accuracy than using unigrams alone.

From all this work, we observe a *performance bottleneck* problem in the polarity classification task: various features have been developed; however, adding more features does not gain much in classification accuracy, and it may even hurt the performance. In our task, although the context and generalization features significantly improve the performance compared to unigrams, we observe a similar *performance bottleneck* problem.

Analysis of the problem The bottleneck problem shows that additional features have much overlap with unigram features, and they may add noise to the classification.

We further analyzed the data, and found that most words in a sentence do not contribute to the classification task. Instead, they can be noise that cannot be removed by adding more features. This could be a crucial reason of the bottleneck discussed above.

To verify this hypothesis, we conducted some experiments on the Medline data set of 2298 sentences used in Section 5.3.2. From each sentence in the data set, we manually extract some words that fully determine the polarity of the sentence. We refer to these words by *extractions* in the following description. For those sentences that do not contain outcomes, nothing is extracted. The following examples are some sentences with different polarity and the extractions from them. These extractions form another data set, which we call the *extraction set*.

Sentence:

Treatment with reperfusion therapies and achievement of TIMI 3 flow are associated with increased short- and medium-term survival after infarction.

Extraction:

increased short- and medium-term survival

Sentence:

In all three studies, a significant decrease in linear growth occurred in children treated with beclomethasone compared to those receiving placebo or non-steroidal asthma therapy.

Extraction:

decrease in linear growth occurred

Sentence:

The doxazosin arm, compared with the chlorthalidone arm, had a higher risk of stroke.

Extraction:

a higher risk of stroke

Sentence:

Prednisolone treatment had no effect on any of the outcome measures.

Extraction:

no effect

Sentence:

There was no significant mortality difference during days 0-35, either among all randomised patients or among the pre-specified subset presenting within 0-6 h of pain onset and with ST elevation on the electrocardiogram in whom fibrinolytic treatment may have most to offer.

Extraction:

no significant mortality difference

We performed the four-way classification task on this extraction set. We constructed UNIGRAMS feature based on the extraction set and used them in the classification. Using 80% of the data as the training data and the rest as the test data, we achieved an accuracy of 93.3%, which is much higher than the accuracy of the four-way classification task on the original sentence set (75.5%).

The fact that we do not extract any words from no-outcome sentences may make the task

easier. Therefore, we removed from the extraction set all sentences that do not contain an outcome, and reran the experiment. This task has three target classes: positive, negative or neutral. We obtained an accuracy of 82.2%. However, performing the three-way classification on the original sentence set only achieves 70.7% accuracy.

The results clearly show that irrelevant words actually introduce a lot of noise in the polarity detection task. Therefore, a new direction of research on the task is to conduct feature selection to remove words that do not contribute to the classification.

A possible solution We took a closer look at the extraction set and found that the extractions usually form a sequence or several sequences in a sentence. Because Hidden Markov Model and Conditional Random Fields are effective models for sequence detection, they will be explored in the future work of this research.

5.5 Summary

In this chapter, we discussed an approach of identifying an inherent property of clinical outcomes – their polarity. Polarity information is important to answer questions related to clinical outcomes. We explored a supervised approach to detect the presence of clinical outcomes and their polarity. We analyzed this problem from various aspects:

- We developed features to represent context information and explored domain knowledge to get generalized features. The results show that adding these features significantly improves the classification accuracy.

- We showed that the feature set has consistent performance on two different text sources, CE and Medline abstracts.

- We evaluated the performance of the feature set on different subtasks of the outcome detection to understand how difficult each subtask is.

- We compared outcome polarity detection to sentiment classification according to different performance of context features on the two tasks. We found that bigram features

have almost no effect on the sentiment classification task, while they improve the classification accuracy of identifying presence and polarity of clinical outcomes.

- We identified a *performance bottleneck* problem in the polarity classification task using a supervised approach. In both the sentiment classification and the outcome polarity detection, we observed that adding more features on top of the unigram features does not lead to major improvement in accuracy. We found a crucial reason for this – the noise in the feature set is not removed by adding more features.

- We proposed to use Hidden Markov Model or Conditional Random Fields to conduct feature selection and thus to remove noise from the feature set.

Chapter 6

Sentence Extraction using Outcome Polarity

As we have addressed in Section 1.5, a crucial characteristic of NFQA is to identify multiple pieces of *relevant* information to construct answers. In the two previous chapters, we discussed properties of semantic classes that are important for detecting the relevance of a piece of information. In this chapter, we investigate the problem of relevance detection using one of the properties: information on the polarity of clinical outcomes, which is discussed in Chapter 5 [Niu et al., 2006].

6.1 Related work

The work most similar to ours is the multi-perspective question answering (MPQA) task, in which Stoyanov et al. [2005] argue that presence of opinions should be identified to find the correct answer for a given question. Some preliminary results are presented to support this claim. Stoyanov et al. [2005] manually created a corpus of opinion and fact questions and answers, OpQA, which consists of 98 documents that appeared in the world press. The documents cover four general topics: President Bush's alternative to the Kyoto protocol; the US annual human rights report; the 2002 coup d'etat in Venezuela; and the 2002 elections in Zimbabwe and Mugabe's reelection. Each topic is covered by between 19 and 33 documents. For each topic, there are 3 to 4 opinion questions, and there are 15 questions in total for all topics.

In their *answer rank experiments*, each sentence in the whole document set is taken as a potential answer to a question. Sentences are first ranked by an information retrieval algorithm based on $tf \cdot idf$ of words in the sentences (step1). Then all fact sentences (sentences that do not express opinions) are removed by some subjectivity filters that distinguish between facts and opinions (step2). In the evaluation, the rank of the first answer to each question in the ranked list after step1 is compared to the rank of the first answer in the list after step2. The mean reciprocal rank (MRR) is used as the evaluation metric. Their results show that the MRR value after step2 is higher than the value after step1.

The results indicate the value of using subjectivity filters in MPQA. The experiment also inspires more thoughts on similar problems. For example, this experiment takes a single sentence as a potential answer, which does not meet very well the needs of drawing on multiple pieces of information in constructing answers to opinion-related questions. Moreover, the strategy of filtering out irrelevant information by removing any sentence that does not express opinions could be too simple for the complex QA task. In our work, we address these problems by exploiting multi-document summarization techniques to find sentences that are relevant/important for answering questions about clinical outcomes, such as "What are the effects of intervention A on disease B?". More specifically, the problem is: after a set of relevant documents has been retrieved, how can we locate constituents of the answer in these documents?[1]

We believe summarization techniques are suitable for our task for two main reasons. First, simply filtering out any information that does not contain an outcome is not appropriate in answer construction. As we discussed in Section 1.1, different outcomes may be present in different patient groups or clinical trials. Therefore, besides information on clinical outcomes, explanation on conditions of the patient groups or the clinical trials can be very important as well. Moreover, not every piece of clinical outcome is important; unimportant outcomes should be discarded. Second, the goal of the summarization task is to find important information with the smallest redundancy, which agrees with that of answer construction in non-factoid QA. The connection between QA and summarization is attracting more attention in the text summarization community. In 2003, the document understanding conferences (DUC) started

[1]This part of the work was carried out in collaboration with Xiaodan Zhu. Xiaodan Zhu participated in the annotation and calculated the score of MMR.

a new task of building short summaries in response to a question. This task was carried on in DUC 2004. In DUC 2005, the intention of modeling "real world complex question answering" is more clear in the *system task*. It is to "synthesize from a set of 25-50 documents a brief, well-organized, fluent answer to meet a need for information that cannot be met by just stating a name, date, quantity, etc" [DUC, 2005]. However, according to our knowledge, summarization techniques have not been explored by current QA systems. In our task, the information needed is the clinical outcomes of an intervention on a disease, and we expect that summarization techniques will help.

On the other hand, we also notice that multi-document summarization cannot replace QA. One important difference between them, as pointed out in [Lin and Demner-Fushman, 2005], is that summaries are compressible in length, i.e., summaries can contain various levels of details, while answers are not. It is difficult to fix the length of answers.

Because of the difference between multi-document summarization and QA, we are not taking the former as the full solution even in the answer generation of a QA task. Instead, we expect that some multi-document summarization techniques can be adapted to the answer generation module in some non-factoid QA tasks. In this chapter, we explore summarization techniques to identify important pieces of information for answer construction.

6.2 *Clinical Evidence* as a benchmark

Evaluation of a multi-document summarization system is difficult, especially in the medical domain where there is no standard annotated corpora available. However, we observe that *Clinical Evidence* (CE) provides a benchmark to evaluate our work against. As mentioned in Section 1.5.1, CE is a publication that reviews and consolidates experimental results for clinical problems; it is updated every six months. Each section in CE covers a particular clinical problem, and is divided into several subsections that summarize the evidence concerning a particular medication (or a class of medications) for the problem, including results of clinical trials on the benefits and harms of the medications. The information sources that CE draws on include medical journal abstracts, review articles, and textbooks. Human experts read the collected information and summarize it to get concise evidence on every specific topic. This is

the process of multi-document summarization. Thus, each subsection of CE can be regarded as a human-written multi-document summary of the literature that it cites.

Moreover, we observed that, generally speaking, the summaries in CE are close to being extracts (as opposed to rewritten abstracts). A citation for each piece of evidence is given explicitly, and it is usually possible to identify the original Medline abstract sentence upon which each sentence of the CE summary is based. Therefore, we were able to create a benchmark for our system by converting the summaries in CE into their corresponding extracted summary. That is, we matched each sentence in the CE summary to the sentence in the Medline abstract on which it was based (if any) by finding the sentence that contained most of the same key concepts mentioned in the CE sentence (this is similar to Goldstein et al. [1999]).

Using CE in our work has an additional advantage. As new results of clinical trials are published fairly quickly, we need to provide the latest information to clinicians. We hope that this work will contribute to semi-automatic construction of summaries for CE.

6.3 Identifying important sentences

6.3.1 Method

We perform summarization at the sentence level, i.e., we extract important sentences from a set of documents to form a summary. For this, we explore a supervised approach. Again, we treat the problem as a classification task, determining whether a sentence is important or not. The same SVM package as in Section 5.3.2 (parameters were set in the same manner) is taken as our machine learning system.

In the classification, each sentence is assigned an importance value by the classifier (SVM)[2] according to a predefined set of features. Sentences with higher values are more important and will be extracted to form a summary of the original documents. Different lengths of summaries (at different compression ratios) are obtained by selecting different numbers of sentences according to their rank in the output of SVMs. In the summaries, the sequence of sentences is

[2]SVM output for each data point in the test set is a signed distance (positive= *important*) from the separating hyperplane. A higher value means that a sentence is more important.

kept the same as in their original documents.

6.3.2 Features to identify important sentences

We use the presence and polarity of an outcome, both manually annotated and determined by the method described in the previous chapter, as features to identify important sentences. In addition, we consider a number of other features that have been shown to be effective in text summarization tasks:

Position of a sentence in an abstract Sentences near the start or end of a text are more likely to be important. We experimented with three different ways of representing sentence position:

1. Absolute position: sentence i receives the value $i - 1$.
2. The value for sentence i is $i/$length-of-the-document (in sentence).
3. A sentence receives a value of 1 if it is at the beginning (first 10%) of a document, a value of 3 if it is at the end (last 10%) of a document, a value of 2 if it is in between.

Sentence length A score reflecting the number of words in a sentence, normalized by the length of the longest sentence in the document [Lin, 1999].

Numerical value A sentence containing numerical values may be more specific and therefore more likely to be important. We tried three options for this feature:

1. Whether or not the sentence contains a numerical value (binary).
2. The number of numerical values in the sentence.
3. Whether or not the sentence contains the symbol '%' (binary).

Maximum Marginal Relevancy (MMR) MMR is a measure of "relevant novelty", and it is formulated using terminologies in information retrieval. Its aim is to find a good balance between relevancy and redundancy. The hypothesis is that information is important if it is both relevant to the topic of interest and least similar to previously selected information, i.e., its

marginal relevance is high. MMR is defined as a linear combination of a relevance measure and a novelty measure [Carbonell and Goldstein, 1998]:

$$MMR = \underset{D_i \in R \setminus S}{\operatorname{argmax}}[\lambda(Sim_1(D_i, Q) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j))]$$

where R is the ranked list of the retrieved documents; Q is a query; S contains a set of documents that have been selected from R, therefore, S is a subset of R; $R \setminus S$ is the set of documents in R that have not been selected; D is a document; $Sim_1(D_i, Q)$ is the similarity between the document D_i and query Q; $Sim_2(D_i, D_j)$ is the similarity between two documents; Sim_1 can be the same as Sim_2. Parameter λ controls the impact of novelty and redundancy in summarization.

We adapt the original definition of MMR to our problem. In our task, R and Q are the same—the list of sentences in all relevant documents from which a summary will be constructed. Because we do not have a specific query set, we set Q to be the same as R, which is often the case in multi-document summarization systems. S is the subset of sentences in R already selected; $R \setminus S$ is the set difference, i.e., the set of sentences in R that are not selected so far; Sim_1 is a similarity metric; and Sim_2 is the same as Sim_1. According to the definition of MMR, when $\lambda = 1$, no redundancy is considered in ranking the sentences, i.e., no sentence will be excluded from the summary because it contains redundant information. When $\lambda = 0$, diversity dominates the constructed summary.

In our experiments, to calculate $Sim(D_i, Q)$, the sentence D_i and the set of documents Q are represented by vectors of $tf \cdot idf$ values ($(1 + tf) \times idf$) of the terms they contain. The similarity is measured by the cosine distance between two vectors. Similarly, we can calculate $Sim(D_i, D_j)$. The score of marginal relevance of a sentence is used as a feature in the experiment (referred to as feature MMR).

6.4 Data Set

The data set in this experiment is the same as in 5.3.2; 197 Medline abstracts cited in 24 subsections (summaries) in CE are used. The average compression ratio of the 24 summaries in CE is 0.25. Out of the total 2298 abstract sentences, 784 contain a clinical outcome (34.1%).

The total number of sentences in the 24 summaries is 546, of which 295 sentences contain a clinical outcome (54.0%). The percentage of sentences containing a clinical outcome in the summaries is larger than in the original Medline abstracts, which matches our intuition that sentences containing clinical outcomes are important.

6.5 Evaluation

In our experiment, we randomly select Medline abstracts that correspond to 21 summaries in CE as the training set, and use the rest of the abstracts (corresponding to 3 summaries in CE) as the test set. The results reported are the average of 50 runs. As the purpose is to observe the behavior of different feature sets, the experimental process can be viewed as a glass box. The system was evaluated by two methods: sentence-level evaluation and ROUGE, an *n*-gram-based evaluation approach. Both of the two methods are commonly used in the summarization community. Randomly selected sentences are taken as baseline summaries.

To evaluate the performance of features, the subsections in CE are viewed as ideal summaries of the abstracts that they cite. The corresponding extraction summaries are used in the sentence-level evaluation, and the original CE summaries are used for ROUGE evaluation.

6.5.1 Sentence-level evaluation

In the experiment, we first observe the performance of using every single feature in the classification. Then, we combine different features and investigate the contribution of the information on clinical outcomes and their polarity in this task.

Comparison of individual features

The precision and recall curves of summaries derived by using every single feature at different compression ratios are plotted in Figure 6.1.

In the figure, the solid horizontal line shows the purely chance performance, which is the baseline. The baseline has a precision of 0.25 because the average compression ratio of CE summaries is 0.25, and the recall at different compression ratios is calculated accordingly.

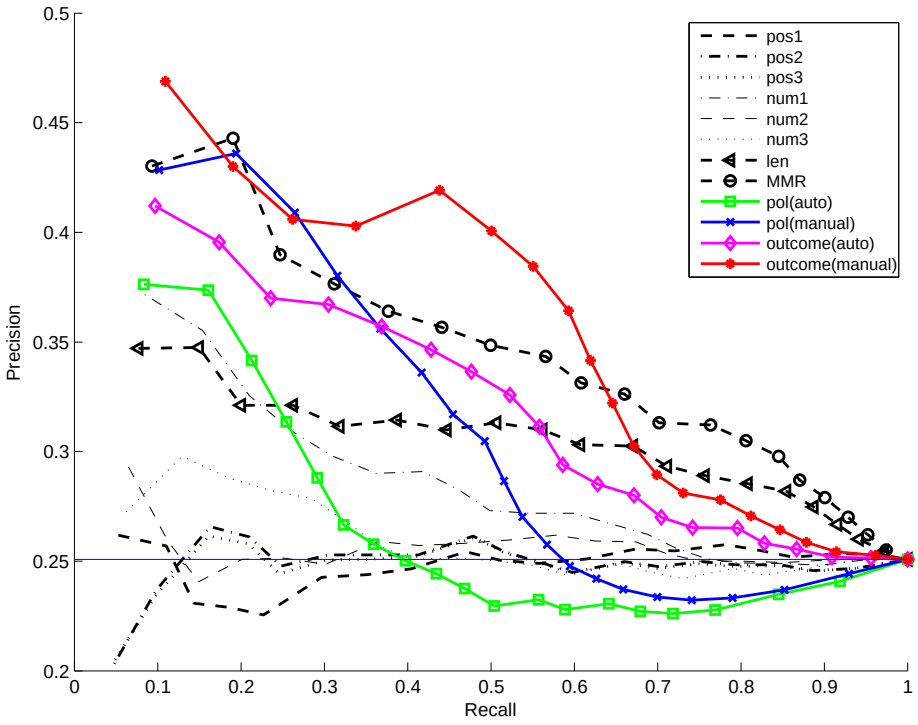


Figure 6.1: Comparison of features

The other four colored solid lines represent the performance of manually or automatically identified clinical outcome and polarity. Although compression ratio is not shown explicitly in the figure, lower compression ratios correspond to lower recall, and compression ratio of 0.25 approximately corresponds to recall of 0.38 on the curve of MMR in the figure. Therefore, the left-hand part of the figure (recall less than 0.5) is more meaningful. Thus, our analysis will focus on this part of the figure.

On the left-hand part of the figure, all four features of the presence and polarity of clinical outcomes are superior to the baseline performance. It is clear that knowledge about clinical outcomes helps in this task. We can see that manually obtained information on presence of outcome outperforms MMR. When compression ratio is relatively low, manually obtained polarity information also performs better than MMR.

For MMR measurement, different values of λ are tested. Higher values produce better summaries, which indicates that relevance is much more important than novelty in this task. The figure shows the best results of MMR ($\lambda = 0.9$). Not surprisingly, MMR is quite effective

Table 6.1: F-score of the summarization with different feature sets in sentence-level evaluation

Compression Ratio	0.1	0.2	0.25	0.3	0.4
Random	.15	.22	.25	.27	.30
MMR	.27	.34	.37	.39	.42
(1)	.26	.36	.40	.42	.44
(2)	**.27**	**.38**	**.41**	**.43**	**.46**
(3)	**.29**	**.40**	**.44**	**.46**	**.48**
(4)	**.27**	**.38**	**.40**	**.43**	**.46**
(5)	**.31**	**.41**	**.44**	**.46**	**.48**

(1): MMR+position+numerical value+length

(2): (1)+automatically identified polarity of clinical outcomes

(3): (1)+manually identified polarity of clinical outcomes

(4): (1)+automatically identified presence of clinical outcomes

(5): (1)+manually identified presence of clinical outcomes

in the task. Other features such as length and numerical value (option 1) also have good effects on the performance.

Combining the features

When features are combined, some of their effects will be additive, and some will cancel out. Table 6.1 shows the F-score of using different combinations of features at different compression ratios. The results of MMR, position, and numerical features listed in the table are the best results obtained (MMR ($\lambda = 0.9$), position (option 2), and numerical features (option 1)). These results are compared to the results of adding information on presence or polarity of clinical outcomes.

As shown in the table, on top of the combination of MMR, position, numerical value, and sentence length features, manually annotated outcome information (either presence or polarity) results in 3 to 5 points of improvement in the F-score. This clearly indicates the importance of the presence/polarity information for the task. Nevertheless, automatically identified informa-

tion regarding presence/polarity improves the performance only slightly when combined with all other features. This suggests the need for developing more accurate techniques for outcome and polarity detection.

The additional benefit from further determining the polarity of the outcome rather than just detecting the presence of an outcome is rather small. Intuitively, we had expected polarity to provide more information on contradicting clinical outcomes and thus to help improve the performance. Looking closely at the data, however, we found that one aspect accounting for the observed result could be that although some sentences are different in polarity, they do not form a contradiction. Rather, they may describe different clinical outcomes. Since some of the outcomes are not important, they are not included in the summaries:

(35) H pylori eradication is highly effective in promoting ulcer healing and preventing subsequent ulcer recurrence.

(36) However, while ulcer symptoms are infrequent during follow-up, a proportion of patients appear to develop gastrooesophageal reflux after eradication.

Example (35) expresses a positive outcome and it is included in the summary. Example (36) describes a negative outcome. Although the polarity in the two sentences is different, they do not contradict each other. It turns out that the second outcome is not important enough to be included in the summary.

In some cases, although the two different outcomes are contradictions, they may not be included in the summary because one or both are not strong evidence, as shown in the following examples:

(37) Results of North American studies of highest methodological quality confirm that H pylori eradication markedly decreases ulcer recurrence.

(38) Nevertheless, 20% of patients in these studies had ulcer recurrence within 6 months, despite successful cure of infection and no reported use of NSAIDs.

Example (38) shows negative evidence on *ulcer recurrence*, while example (37) shows positive evidence on it. However, (37) is strong evidence while (38) is not. Hence, only the first is included in the summary. As a result, further determining whether a clinical outcome is

Table 6.2: Comparison of feature sets in single summary at compression ratio 0.25

(1) : MMR+position+numerical value+length

	$>$ (1)	= (1)	$<$ (1)
(1)+Polarity(auto)	10	7	7
(1)+Polarity(manual)	15	6	3
(1)+Outcome(auto)	11	6	7
(1)+Outcome(manual)	15	2	7

positive, negative, or neutral does not have provide more evidence to support the importance of a sentence.

We also observed the performance of different combinations of features on every single summary at compression ratio 0.25, and show the results of feature sets with and without presence or polarity information in Table 6.2. The numbers in the table are numbers of summaries. For example, the number in the second row, the first column is 10, which means that in 10 summaries the performance (F-score) of using feature set (1)+Polarity(auto) is better than using feature set (1). The complete result is plotted in Appendix J.

6.5.2 ROUGE

As an alternative evaluation, we use the ISI ROUGE package [Lin, 2004], which compares a summary generated by a text summarization system with a benchmark summary by considering overlapping units such as *n*-grams, word sequences, and word pairs (word sequences and word pairs allow gaps between words). Our evaluation was carried out with various ROUGE parameters. Unlike the sentence-level evaluation, the results showed little difference in the performance of different combinations of features. Table 6.3 shows the ROUGE-L score of three feature sets. ROUGE-L is a measure of the longest common subsequence in the two texts to be compared, where the longest common subsequence of two sequences X and Y is a common subsequence with maximum length. For example, let $X = [\text{there are } \textbf{four cats}]$, $Y_1 = [\text{I have } \textbf{four cats}]$ and $Y_2 = [\textbf{four} \text{ cute } \textbf{cats} \text{ are playing with a ball}]$, then ROUGE-

Table 6.3: ROUGE-L score of different feature sets

Compression Ratio	0.1			0.2			0.3			0.4		
	P	R	F	P	R	F	P	R	F	P	R	F
MMR	.46	.18	.25	.40	.31	.33	.35	.40	.36	.30	.45	.35
(1)	.46	.18	.25	.41	.31	.34	.35	.40	.36	.30	.45	.35
(2)	.46	.18	.25	.41	.31	.34	.35	.39	.36	.30	.46	.35

(1): MMR+position+numerical value+length

(2): automatically identified presence of clinical outcomes

L(X, Y_1) = ROUGE-L(X, Y_2).

As the table shows, adding position, numerical value, and length features to MMR does not improve the F-score compared to using MMR alone. Furthermore, adding automatically identified presence of outcomes does not make any difference in F-score. One reason for the result that different combinations of features perform almost the same in ROUGE evaluation could be that it is difficult for an overlap-based metric to capture the difference if the content of two sets is similar. For example, only a small difference might be measured by ROUGE when comparing the inclusion of both a positive and a negative clinical outcome of an intervention in treatment of a disease in the summary with the inclusion of only one of them.

6.6 Summary

In this chapter, we proposed an approach of identifying relevant pieces of information in answer construction for a specific task, i.e., answering questions about clinical outcomes. We explored multi-document summarization techniques to rank sentences according to their importance in the context of constructing answers in QA. Our hypothesis is that information on presence or polarity of clinical outcomes in a sentence would help identify answer components. Such information is used as features, together with some other features that have been shown effective in previous summarization tasks. We investigated every single feature and show that in general, the performance of the summarization system using outcome informa-

tion is superior to the baseline, especially when the compression ratio is relatively low (around the average compression ratio of the CE summaries). This result demonstrates the effectiveness of such information. We also found that when features are combined, manually annotated presence/polarity information improves the performance achieved by using all other features. This further reveals the value of such information in the task. However, using automatically detected presence/polarity information results in only a slight improvement. Thus, our next step will be to build a more accurate outcome information detection system. Another finding is that there is almost no difference between presence and polarity of clinical outcomes in the task. Nevertheless, polarity information is mandatory for answering some questions, such as questions about side effects of interventions.

In this experimental setting, the questions are about general clinical outcomes such as "What is the effect of intervention A on disease B?". For more specific questions as in Chapter 4, finer-grained analysis needs to be performed to find the answer, e.g., identifying cores of semantic classes. For any case, text summarization techniques that address the problem of finding relevant information and avoiding redundancy, e.g., MMR and various features, can be used to identify answer components in NFQA.

Chapter 7

Conclusion

In the research that this dissertation presents, we have analyzed characteristics of NFQA. We found that answers to non-factoid questions are usually more complex than named entities, and multiple pieces of information are often needed to construct a complete answer. We proposed a novel approach to address these characteristics. Important subtasks in different components of the new approach were identified, and automatic methods were developed to solve the problems.

7.1 Summary of contributions

The contributions of this research have been presented in Section 1.6. We summarize them here, adding emphasis on evaluation results.

The QA approach based on semantic class analysis We use frames to represent semantic classes in scenarios and proposed an approach taking semantic class analysis as the organizing principle to answer non-factoid questions. This approach contains four major components:

- Detecting semantic classes in questions and answer sources
- Identifying properties of semantic classes
- Question-answer matching: exploring properties of semantic classes to find relevant pieces of information

- Constructing answers by merging or synthesizing relevant information using relations between semantic classes

We investigated NFQA in the context of clinical question answering, and focused on three semantic classes that correspond to roles in the commonly accepted PICO format of describing clinical scenarios. The three classes are: the problem of the patient, the intervention used to treat the problem, and the clinical outcome.

Extracting semantic classes and analyzing their relations We used rule-based approaches to identify clinical outcomes and relations between instances of interventions in sentences. In the combination approach of outcome identification, a set of cue words that signal the occurrence of an outcome were collected and classified according to their PoS tags. For each PoS category, the syntactic components it suggests were summarized to derive rules of identifying boundaries of outcomes. This approach can potentially be applied to identify or extract other semantic classes. We identified six common relationships between different instances of interventions in a sentence and developed a cue-word-based approach to identify the relations automatically. These approaches were evaluated on a small data set annotated by human judges.

Identifying cores of semantic classes We showed how cores of interventions, problems, and outcomes in a sentence can be identified automatically by developing an approach exploring semi-supervised learning techniques. The evaluation shows that each type of feature used in the approach made contributions to the classification, and they are complementary in the task. The structure of the data is followed by the semi-supervised model. Therefore, unlabeled data is effectively exploited in the classification.

Detecting polarity of clinical outcomes We developed a method using a supervised learning model to automatically detect polarity of clinical outcomes. The results show that context features and category features significantly improved classification accuracy compared to using unigrams alone. This method has stable performance on different sources of medical text. We also identified a cause of the bottleneck of performance of supervised approach in polarity

detection, and showed evidence by the results of manual experiments.

Extracting components for answers We built explicit connection between text summarization and answer construction in NFQA, i.e., both of them need to identify important information and avoid redundancy. We constructed a summarization system that explored a supervised classification model to extract important sentences for answer construction. We investigated the role of presence and polarity of clinical outcomes in this task. The evaluation shows that presence/polarity of clinical outcomes helps the summarization. However, accuracy of automatic approaches is not high enough to make a substantial improvement in the performance. An additional advantage of the polarity information is that it is mandatory when answering questions about benefits or harms of interventions.

Generalization of the approach of semantic class analysis Our approach analyzes semantic classes involved in scenarios to find answers to non-factual questions. The importance of semantic classes has been shown both in theory of linguistics and by the success of IE systems. The properties of semantic classes that we identified for the medical domain apply to other domains as well. A core is the essence of an instance of a semantic class; hence, instances of any semantic class have cores. Polarity is a property for any semantic class that can be evaluated as *good*, *bad*, or *neutral*, although most of work on polarity analysis focuses on opinions. In each subproblem of analysis of semantic classes discussed in the thesis, we use general rule-based or machine learning approaches, although some domain-specific features are incorporated in some tasks. Therefore, we expect that our approach can be used to answer non-factual questions in other domains as well.

7.2 Future work

7.2.1 Extensions

Short-term future work includes overcoming some limitations or extending the current work.

- Refine rules of using adjectives to identify clinical outcomes. In both detecting the occurrence of outcomes and determining the boundaries, the usage of adjective cue

words is harder to describe as compared to nouns and verbs. Finer-grained rules will be able to handle these cases better.

- Extend rules of analyzing relations between instances of interventions to deal with ambiguity. Although ambiguous cue words found in the relation analysis indicate one relation more often than the other, it will be helpful to have more specific rules that disambiguate the occurrence of such a cue.

- Investigate approaches to automatically collect words that express *more* or *less*, *good* or *bad* to generate the *change phrase* features in detecting polarity of clinical outcomes.

- Explore more sophisticated techniques to handle negations in polarity detection. Negation is a very complex linguistics phenomenon. In-depth analysis of negations, for example, categorizing expressions of negations and investigating the syntactic relations of the expressions, can be performed to reduce errors.

- Perform feature selection to remove redundant features in the polarity detection task.

- Improve boundary detection of the core identification task. In our current work, detecting boundaries of cores uses the UMLS knowledge base. Since such a knowledge base does not exist for some domains, machine learning approaches can be explored to improve boundary detection by investigating syntactic constituents of cores.

- Use cores of semantic classes to identify answer components. Various strategies can be taken in calculating the similarity of a question and a piece of information in the answer source. For example, only cores may be taken into account, or the complete description may be considered while giving more weights to cores.

- Explore unsupervised summarization techniques to extract answer components. Unsupervised methods are important when annotated data is not available.

7.2.2 Directions for future work

On the basis of our research in non-factoid QA, we envision several major directions for future work.

Improving polarity classification Polarity is a property of many things, such as clinical outcomes and opinions. Automatic polarity classification, as shown in this thesis and in the research on sentiment analysis, is a crucial issue in many applications including QA. To improve polarity classification, we found three research problems of great interest:

- The granularity of polarity analysis. As described in Section 5.1, current polarity research is at three levels: words, sentences, and documents. Determining polarity of words will contribute to higher-level analysis. The advantage of performing sentence-level analysis is that a sentence often expresses a polarity that is rather complete and independent. Nevertheless, the difficulty at this level is that a sentence may contain more than one polarity unit and they may have different polarity. In addition, the polarity of a sentence may be related to its adjacent sentences. For example, a sentence may continue the same positive/negative polarity as its previous sentence although its own expression is neutral. Document-level analysis can be too coarse-grained. Take the main research at the document-level, movie reviews, as an example. A review usually contains discussions on both positive and negative aspects of a movie. Although the polarity of the whole document may show the general view of a reviewer, it will be much more informative to identify what in the movie is successful and what is not in the reviewer's opinion. This will rely on sentence-level analysis. Therefore, we believe it is important to investigate the difficulty of sentence-level analysis mentioned above, and develop new approaches to address it.

- Feature selection in polarity classification. As discussed in Section 5.4, polarity of a clinical outcome is often determined by a sequence of words in a sentence. It will be interesting to see if the same observation can be obtained from other polarity detection tasks, such as sentence-level sentiment classification. Current work in the opinion domain often uses the semantic orientation of a set of words to detect polarity. This may indicate the existence of a set of words that fully determine the polarity. This observation may also hold at the document level. Because the goal of document-level sentiment classification is to get the general view of the reviewer, it may not be necessary to consider all words or polarity expressions in the document, as it is in current

work. Instead, the polarity of the document may be set by one or a small amount of sentences.

- The role of in-depth semantic analysis in polarity classification. As pointed out in Pang et al. [2002], the task of sentiment classification is more difficult than the traditional topic categorization task. Some examples in the paper show that in sentiment classification, even if a sentence or a document contains a lot of positive expressions, it may turn out to be negative. This suggests that semantics of languages play a more important role in the polarity classification task compared to the topic categorization task. However, dominant features in current approaches for polarity classification are similar as in topic categorization, i.e., using bag-of-words. Therefore, an important research direction is to effectively incorporate appropriate features capturing semantics in text, such as negations.

Using topic detection to find answer components In Chapter 6, we described the approach of using multi-document summarization techniques to identify relevant pieces of information in a set of relevant documents to construct the answer. It will be interesting to further investigate this approach in the case that a relevant document discusses more diverse topics. For such documents, it may be useful to first identify which part in a document is about the topic addressed by the question, and then to apply summarization techniques to extract important information from that particular part. This will require detecting topics in a document.

The problem of topic detection has been investigated for text summarization [Nomoto and Matsumoto, 2001, 2003; Hardy et al., 2002]. The idea is to detect major topics discussed in a document, and then extract important information from each topic to compose a summary. This approach addresses the importance and redundancy by selecting salient information from diverse topics. Another related research area is topic segmentation [Caillet et al., 2004], which recently is explored mostly in speech, dialogue, and news [Purver et al., 2006; Malioutov and Barzilay, 2006; Rosenberg and Hirschberg, 2006].

We will use topic areas (a topic area is a segment of text discussing one topic) to find the answer to a given question. Therefore, after topic areas in a document are identified, we need a similarity measure to select the topic that is most similar to the question. Then, we will extract

important information from this topic area. In QA, we usually have a set of documents that are relevant to the question. Hence, this process will be conducted for every document, and the answer will be constructed on the basis of the information extracted from each document.

Exploring textual context of frames In our frame-based semantic class analysis approach, an answer frame in a document will be matched to the question frame by comparing the corresponding semantic classes in the two frames. Then, information in the answer frame will be extracted from the document to construct the answer. However, it may not be appropriate to only extract the answer frame as it is usually not isolated from its textual context in the document. Instead, this context often describes related information such as the background, the preconditions, and the explanation of the answer frame. Missing this information can result in misleading answers. Therefore, another research direction of locating relevant pieces of information as answer components is to explore discourse analysis.

One of the most popular discourse theories is the Rhetorical Structure Theory proposed by Mann and Thompson [1986]. The central notion in the theory is that of *rhetorical relation*, which is a relation that holds between two non-overlapping text spans. Some examples of rhetorical relations are: *justification*, *equivalence*, *contrast*, *cause*, *condition*, and *change-topic*. The text spans can be text sequences within a sentence [Marcu and Echihabi, 2002], or they can be sentences [Saito et al., 2006].

We can perform discourse analysis to understand how sentences in the context of an answer frame are related to each other, i.e., if the sentences are connected by some **important** rhetorical relations. The set of rhetorical relations that are important in this task can be collected manually or via some supervised learning approaches. To construct the answer, both the answer frame and its important context determined by the rhetorical relations will be extracted from the original document.

Cross-sentence scenario analysis Currently, our work is at the sentence level, i.e., slot fillers of a frame are extracted from a single sentence. Consequently, if a sentence only contains the instances of some semantic classes in a frame, the frame extracted from the sentence will have some empty slots. Because empty slots may cause confusion in the matching process, we need to resolve them by locating the correct instances in the document.

We identified two causes of empty slots. In one case, instances of a semantic class are simply omitted in a sentence because they are described in the previous sentence(s). In such case, the meaning of this sentence is easy to interpret by taking into account the previous sentence(s). Hence, it is easy to fill the empty slots by examining the corresponding semantic classes in the previous sentence(s). In the other case, in a sentence, some instances of semantic classes are referred to by various expressions, e.g., pronouns. One way to address this case is to explore coreference resolution to locate the correct slot fillers.

Coreference resolution is a major research area in computational linguistics. Coreference can be tackled by exploring syntactic constraints, semantics, and discourse information. Recently, statistical approaches using various models have been developed to improve the accuracy of coreference resolution [Bergsma and Lin, 2006; Yang et al., 2006]. We need to examine current approaches in detail for possible solutions to our problem of resolving empty slots. This problem, however, may present new features that require new approaches to coreference resolution.

Appendix A

List of abbreviations

CE	Clinical Evidence
DUC	Document Understanding Conferences
EBM	Evidence-based Medicine
EBOC	Evidence-based On Call
EPoCare	Evidence at the Point of Care
FBQA	Fact-based Question Answering
HITIQA	High-Quality Interactive Question Answering
IE	Information Extraction
IR	Information Retrieval
LLE	Locally Linear Embedding
MMR	Maximum Marginal Relevancy
MPQA	Multi-Perspective Question Answering
MRR	Mean Reciprocal Rank
MUC	Message Understanding Conference
NE	Named Entities
NFQA	Non-factoid Question Answering
QA	Question Answering
RBF	Radial Basis Function
SO	Semantic Orientation
SVM	Support-Vector Machine
UMLS	Unified Medical Language System

Appendix B

A subset of syntactic tags in Apple Pie Parser

PoS tags:

CC: Coordinating conjunction; DT: Determiner; IN: Preposition or subordinating conjunction; JJ: Adjective; JJR: Adjective, comparative; NN: Noun, singular or mass; NNP: Proper noun, singular; NNS: Noun, plural; NNPS: Proper noun, plural; NNPX: NNP + NNPS; RB: Adverb; TO: to; VB: Verb, base form; VBN: Verb, past participle; VBZ: Verb, 3rd person singular present

Phrase tags:

ADVP: Adverb phrase; NP: Noun phrase; NPL: NP which has no NP in its decendents, lowest NP; PP: Prepositional phrase; VP: Verb phrase; S: Sentence

Appendix C

The effect of σ in the RBF kernel in core identification

This appendix shows the classification results using different σ values in the RBF kernel (cosine distance is used here). Figure C.1 shows that when σ is in a reasonable range, the performance of the classification is not sensitive to its change.

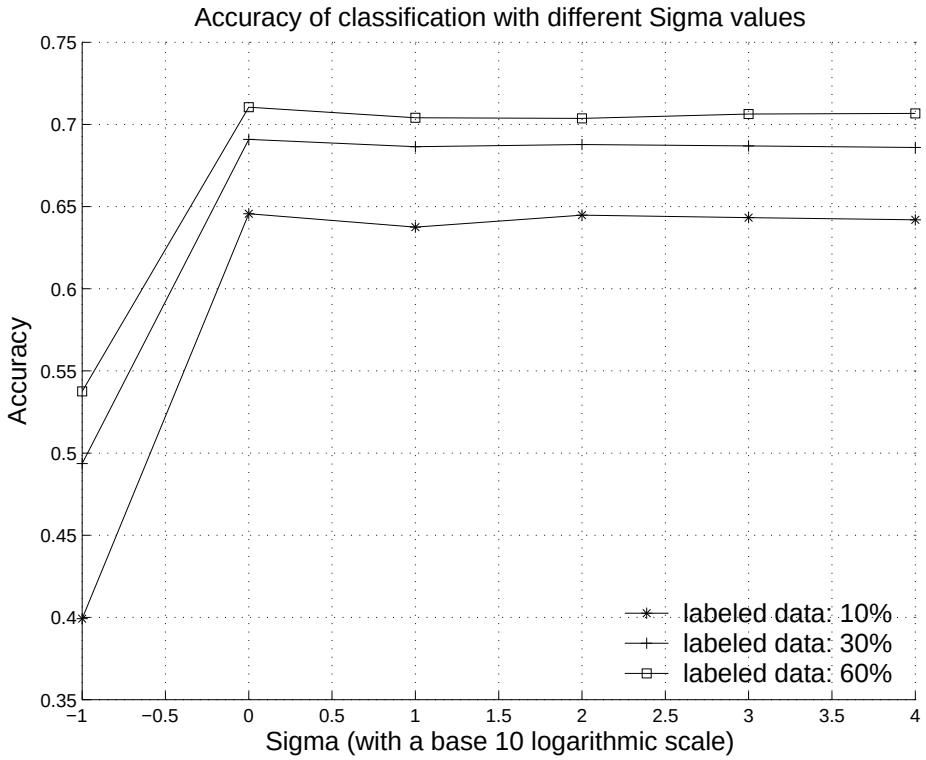


Figure C.1: Classification results with different values for σ

Appendix D

Algorithm for Boundary Detection of Clinical Outcomes

Input: $S = \{S_1, \cdots S_n\}$: a set of parsed sentences.

Output: $O = \{O_1, \cdots O_n\}$: clinical outcomes identified from S.

Notations: $[l_P, r_P]$ — the boundary of a word sequence P.

$NP(w)$ — the innermost noun phrase containing the word w.

$AP(w)$ — the innermost adjective phrase containing w.

$C(w)$ — the clause containing w.

$com([l_1, r_1], [l_2, r_2])$ — combination of intersecting intervals; return $[min(l_1, l_2), max(r_1, r_2)]$.

$max([l_1, r_1], [l_2, r_2])$ — maximum cover; return the larger interval.

1: For each sentence S_i in S:

2: Initialize the boundary set B; set the position counter pc to the start of S_i.

3: While pc does not reach the end of S_i:

4: get next word w.

5: compute the extraction boundary for the following cases of w:

6: (i) $w \in$ [*difference*, *superior*, *effective*]:

7: (a) identify $NP(w)$.

8: (b) identify the propositional phrase PP which immediately follows $NP(w)$.

9: (c) record the extraction boundary $[l_{NP(w)}, r_{PP}]$ in B.

(ii) w is a cue verb:

(a) check negation of w and label the negative status on w.

(b1) if w is active, then

identify E which is the noun phrase immediately following w.

(b2) if w is passive, then

identify E which is the word sequence between the start of $C(w)$ and w.

(c) record the boundary $[l_E, r_E]$ in B.

(iii) w is a cue noun:

(a) identify $NP(w)$.

(b) identify the phrase $\hat{P}$ which contains w and is one level higher than $NP(w)$.

(c) if $\hat{P}$ is a noun phrase, then record $[l_{\hat{P}}, r_{\hat{P}}]$ in B;

else, record $[l_{NP(w)}, r_{NP(w)}]$ in B.

(iv) w is an adjective:

(a) identify $NP(w)$.

(b) identify $AP(w)$.

(c) record $max([l_{NP(w)}, r_{NP(w)}], [l_{AP(w)}, r_{AP(w)}])$ in B,

EndOfWhileLoop.

Combine boundaries in B using $combine$ function until there are no intersecting boundaries.

Output outcome O_i – word sequences extracted from S_i as indicated by the boundaries in B.

EndOfForLoop.

Appendix E

Sample output of MetaMap

Sentence:

It found that the combined rate of myocardial infarction, stroke, or death was slightly lower in the lower dose than in the higher dose groups at 3 months.

Output of MetaMap:

Phrase: “It”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “found”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “that”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “the”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “combined”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “rate”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “of myocardial infarction”

Meta Candidates (6)

1000 Myocardial Infarction [Disease or Syndrome]

861 Infarction [Finding,Pathologic Function]

861 Myocardial [Functional Concept]

805 MI <2> (Without) [Qualitative Concept]

789 MIS (Mullerian duct inhibiting substance) [Amino Acid, Peptide, or Protein,Hormone]

789 Myocardium [Tissue]

Meta Mapping (1000)

1000 Myocardial Infarction [Disease or Syndrome]

Phrase: “stroke”

Meta Candidates (1)

1000 Stroke (Cerebrovascular accident) [Disease or Syndrome]

Meta Mapping (1000)

1000 Stroke (Cerebrovascular accident) [Disease or Syndrome]

Phrase: “or”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “death”

Meta Candidates (3)

1000 Death <1> (Cessation of life) [Finding,Organism Function]

916 LIQUEFACTION [Laboratory or Test Result]

900 Expired [Functional Concept]

Meta Mapping (1000)

1000 Death <1> (Cessation of life) [Finding,Organism Function]

Phrase: “was”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “slightly lower”

Meta Candidates (4)

861 Lower [Functional Concept]

805 LO [Pharmacologic Substance]

694 SLIGHTLY [Idea or Concept]

623 Slight [Qualitative Concept]

Meta Mapping (888)

694 SLIGHTLY [Idea or Concept]

861 Lower [Functional Concept]

Phrase: “in the lower dose”

Meta Candidates (3)

789 dosage (Dosages) [Quantitative Concept]

694 Lower [Functional Concept]

638 LO [Pharmacologic Substance]

Meta Mapping (853)

694 Lower [Functional Concept]

789 dosage (Dosages) [Quantitative Concept]

Phrase: “than”

Meta Candidates (0): <none>

Meta Mappings: <none>

Phrase: “in the higher dose groups”

Meta Candidates (5)

827 Groups [Intellectual Product]

701 High dose [Quantitative Concept]

627 High [Qualitative Concept]

627 High <1> (Euphoric mood) [Mental Process]

589 dosage (Dosages) [Quantitative Concept]

Meta Mapping (879)

701 High dose [Quantitative Concept]

827 Groups [Intellectual Product]

Phrase: “at 3 months”

Meta Candidates (1)

861 months (month) [Temporal Concept]

Meta Mapping (861)

861 months (month) [Temporal Concept]

Appendix F

Sample output of Minipar

This appendix shows the output of the Minipar parser on a sentence. The output of the parser is used to construct the *syntactic relations* features in the core identification task.

Sentence:

Thrombolysis reduces the risk of dependency, but increases the chance of death.

Candidates of cores:

thrombolysis, dependency, death

Output of Minipar:

(

E0 (() fin C *)

1 (Thrombolysis N 2 s (gov reduce))

2 (reduces reduce V E0 i (gov fin))

E2 (() Thrombolysis N 2 subj (gov reduce) (antecedent 1))

3 (the Det 4 det (gov risk))

4 (risk N 2 obj (gov reduce))

5 (of Prep 4 mod (gov risk))

6 (dependency N 5 pcomp-n (gov of))

7 (, U 2 punc (gov reduce))

8 (but U 2 punc (gov reduce))

9 (increases increase V 2 conj (gov reduce))

E3 (() Thrombolysis N 9 subj (gov increase) (antecedent 1))

10 (the Det 11 det (gov chance))

11 (chance N 9 obj (gov increase))

12 (of Prep 11 mod (gov chance))

13 (death N 12 pcomp-n (gov of))

14 (. U * punc)

)

Relations:

(thrombolysis subj–of increase), (thrombolysis subj–of reduce)

(dependency pcomp-n–of of)

(death pcomp-n–of of)

Figure F.1: Example of dependency triples extracted from output of Minipar parser.

Appendix G

List of words for building CHANGE PHRASES features

This appendix shows the list of **good**, **bad**, **more**, and **less** words collected from CE in detecting polarity of clinical outcomes.

Good:

cured	vitality	relaxing	benefit	tolerability	improvement
right	effective	stable	best	better	pleasurable
relaxation	favour	beneficial	safety	prevents	successful
satisfaction	significant	superior	contributions	reliability	robust
tolerated	improving	survival	favourable	reliable	recovered
judiciously	consciousness	efficacy	prevented	satisfied	prevent
advantage	encouraging	tolerance	success	significance	improved
improves	improve	improvements			

Bad:

depression	acute	sore	outpatient	disabling	diabetes
difficulties	dysfunction	distorted	poorer	unable	prolonged
irritation	disruptive	pathological	mutations	disease	infection

harms	difficulty	weakened	inactive	stressors	hypertension
adverse	insomnia	relapsing	malignant	suffer	exacerbate
dryness	fever	overestimate	constipation	deposition	colic
tension	hazards	diarrhoea	weakness	irritability	insidious
distress	weak	cancer	emergency	risk	block
unsatisfactory	blinding	nausea	traumatic	wound	intention
loses	intensive	relapse	recurrent	extension	die
cancers	malaise	crying	toxic	injury	confounding
complaints	misuse	insignificant	poisoning	anoxic	amputation
death	nightmares	deteriorate	fatal	injuries	fatigue
invasive	suicide	chronic	relapsed	disturbances	confusion
died	fluctuating	severities	delusions	compulsions	conflict
trauma	cried	impair	severe	tremor	weaker
illness	inpatients	worry	rebound	worse	reversible
dizziness	attacks	pointless	disorders	dyskinesia	risks
fatty	negative	conflicting	upset	fishy	hard
harm	bleeding	inflammatory	hampered	underpowered	obstruction
headache	problem	bleeds	panic	loss	odds
retardation	dysfunctional	render	difficult	drowsiness	lack
suicidal	obsessions	impaired	cough	severity	suffering
violent	strokes	virus	stroke	flatulence	fibrates
blind	burning	faintness	suffered	threatening	misdiagnosing
bitter	excessive	diabetics	malfunction	abnormal	deterioration
bad	confounded	sadness	mortality	disturbance	agitated
attack	infections	negativistic	deaths	poor	wrong
worsening	adversely	insufficient	scarring	headaches	disability
overdose	serious	delayed	discomfort	sweating	morbidity
nerve	parkinson	toxicity	nervous	pain	stress

weakens	incorrect	disorder	worsened	malformations	blinded
rigidity	prolong	adversity	abuse	lacked	dyspepsia
sads	onset	failure	inadequate	sensitivity	impairment
dementia	harmful				

Increase:

increase enhance elevation higher exceed enhancement peaked more excess

Decrease:

below lower decrease fall low reduce decline less little mild drop fewer

Appendix H

Using an RBF kernel in detecting polarity of clinical outcomes

This appendix shows the accuracy of using a non-linear kernel, RBF ($exp(-d^2(x_i, x_j)/\sigma^2)$), in SVMs in the task of detecting positive and negative clinical outcomes in CE text. Using the feature set of presence of combining UNIGRAMS with DISEASE and MORE/LESS, the accuracy of the classification obtained with different σ values is shown in Table H.1. We can see that small σ stretches the distance between data points and makes the classification very difficult. When σ is larger, the performance is not very sensitive to its change, and becomes relatively stable.

Table H.1: Results of positive/negative classification using an RBF kernel

σ^2	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}	10^{5}
Accuracy (%)	34.9	34.9	34.9	87.9	91.9	92.3	92.7	92.7

Appendix I

Results of the summarization with different feature sets in sentence-level evaluation

Table I.1: Results of the summarization with different feature sets in sentence-level evaluation

Compression Ratio	0.1			0.2			0.25			0.3			0.4		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Random	.25	.11	.15	.25	.20	.22	.25	.26	.25	.25	.31	.27	.25	.40	.30
MMR	.44	.19	.27	.38	.31	.34	.36	.38	.37	.36	.44	.39	.34	.57	.42
(1)	.44	.19	.26	.40	.33	.36	.40	.41	.40	.38	.48	.42	. 36	.58	.44
(2)	.45	.20	**.27**	.42	.35	**.38**	.40	.42	**.41**	.39	.49	**.43**	.37	.61	**.46**
(3)	.45	.20	**.27**	.41	.35	**.38**	.40	.42	**.40**	.39	.48	**.43**	.37	.61	**.46**
(4)	.49	.21	**.29**	.44	.38	**.40**	.43	.46	**.44**	.41	.52	**.46**	.38	.64	**.48**
(5)	.51	.22	**.31**	.45	.38	**.41**	.43	.46	**.44**	.42	.53	**.46**	.39	.65	**.48**

(1): MMR+position+numerical value+length

(2): (1)+automatically identified polarity of clinical outcomes

(3): (1)+manually identified polarity of clinical outcomes

(4): (1)+automatically identified presence of clinical outcomes

(5): (1)+manually identified presence of clinical outcomes

Appendix J

F-score of combinations of features in each single summary

The F-score of each summary at compression ratio 0.25 is presented in Figure J.1 to observe the performance of different combinations of features on every single summary. The diagram at the top shows the results of features including presence of outcomes, and the one on the bottom shows the results of features including polarity of outcomes.

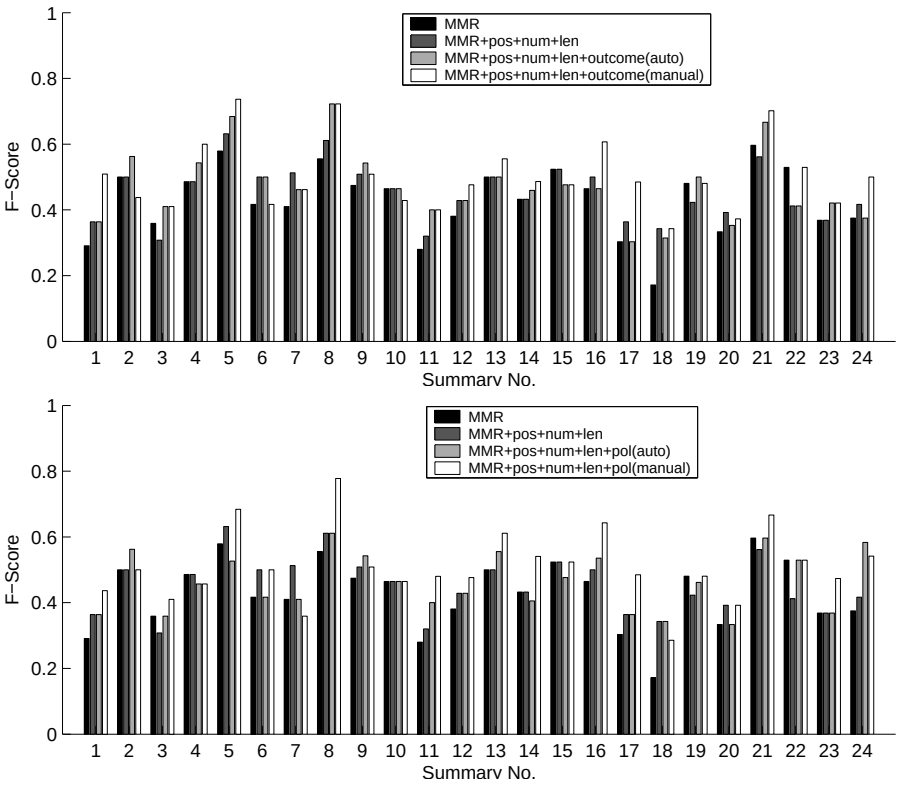


Figure J.1: The performance of different combinations of features in each summary

Bibliography

Alpha, S., Dixon, P., Liao, C., and Yang, C. (2001). Oracle at TREC 10: Filtering and question–answering. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*.

Ananiadou, S. and Tsujii, J., editors (2003). *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*. Association for Computational Linguistics (ACL), PA, USA.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the American Medical Informatics Association Symposium*, pages 17–21.

Baker, C. F., Fillmore, C. J., and Cronin, B. (2003). The structure of the Framenet database. *International Journal of Lexicography*, 16(3):281–296.

Ball, C. M. and Phillips, R. S. (2001). *Evidence-based on Call: Acute Medicine*. Harcourt Publishers Limited, London.

Barbosa, D., Barta, A., Mendelzon, A., Mihaila, G., Rizzolo, F., and Rodriguez-Gianolli, P. (2001). Tox — the toronto xml engine. In *Proceedings of the International Workshop on Information Integration on the Web, Rio de Janeiro, 2001*.

Barton, S., editor (2002). *Clinical Evidence*. BMJ Publishing Group, London.

Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. (2000). Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23th International Conference on Research and Development in Information Retrieval (SIGIR- 2000)*, pages 192–199.

Bergsma, S. and Lin, D. (2006). Bootstrapping path-based pronoun resolution. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100.

Brill, E. (1993). *A Corpus-Based Approach to Language Learning (PhD thesis)*. University of Pennsylvania, PA, USA.

Caillet, M., Pessiot, J. F., Amini, M. R., and Gallinari, P. (2004). Unsupervised learning with term clustering for thematic segmentation of texts. In *Proceedings of Recherche d'Information Assistée par Ordinateur (RIAO)*.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Cardie, C., Wiebe, J., Wilson, T., , and Litman, D. (2003). Combining low-level and summary representations of opinions for multi-perspective question answering. In *Proceedings of the AAAI Spring Symposium: New Directions in Question Answering*, pages 20–27.

Català, N., Castell, N., and Martín, M. (2003). A portable method for acquiring information extraction patterns without annotated corpora. *Natural Language Engineering*, 9(2):151–179.

Chang, C.-C. and Lin, C.-J. (2001). LIBSVM–a library for support vector machines. In *http://www.csie.ntu.edu.tw/ cjlin/libsvm/*.

Cheng, G. Y. (2004). A study of clinical questions posed by hospital clinicians. *J Med Libr Assoc.*, 92(4):445–458.

Chieu, H. L. and Ng, H. T. (2003). Named entity recognition with a maximum entropy approach. In *Proceedings of 7th Conference on Computational Natural Language Learning*, pages 160–163.

Cimino, J. J. (1996). Linking patient information systems to bibliographic resources. *Methods of Information in Medicine*, 35(2):122–126.

Clarke, C. L. A., Cormack, G. V., Kisman, D. I. E., and Lynam, T. R. (2000). Question answering by passage selection. In *Proceedings of the Ninth Text Retrieval Conference (TREC 2000)*.

Clarke, C. L. A., Cormack, G. V., and Lynam, T. R. (2001). Exploiting redundancy in question answering. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA (SIGIR-2001)*, pages 358–365.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Cucerzan, S. and Yarowsky, D. (1999). Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Diekema, Yilmazel, A., Chen, O., Harwell, J., Sarah, H., L., , Liddy, and D., E. (2003). What do you mean? finding answers to complex questions. In *Proceedings of the AAAI Spring Symposium: New Directions in Question Answering*.

Dowty, D. R. (1991). Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

DUC (2005). Document understanding conferences. In *http://duc.nist.gov/duc2005*.

Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. (2002). Web question answering: is more always better? In *Proceedings of the 25th International Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland (SIGIR-2002)*, pages 291–298.

Ebell, M. H. (1999). Information at the point of care: answering clinical questions. *J AM Board Fam Pract.*, 12(3):225–235.

Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Monceaux, L., Robba, I., and Vilnat, A. (2001). Finding an answer based on the recognition of the question focus. In *Proceedings of the Tenth Text Retrieval Conference, National Institute of Standards and Technology (NIST), Nov. 13-16, Gaithersburg, Maryland, USA (TREC 2001)*, pages 362–370.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280:20–32.

Florian, R., Ittycheriah, A., Jing, H., and Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of 7th Conference on Computational Natural Language Learning*, pages 168–171.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Goldstein, J., Kantrowitz, M., Mittal, V., and Carbonell, J. (1999). Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128.

Gorman, P., Ash, J., and Wykoff, L. (1994). Can primary care physicians' questions be answered using the medical journal literature? *Bulletin of Medical Library Association*, 82(2):140–146.

Harabagiu, S., Maiorano, S., Moschitti, A., and Bejan, C. (2004). Intentions, implicatures and processing of complex questions. In *Workshop on Pragmatics of Question Answering, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 31–42.

Harabagiu, S. M., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V., and Morărescu, P. (2001a). The role of lexico-semantic feedback in open-

domain textual question–answering. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 274–281.

Harabagiu, S. M., Moldovan, D. I., Paşca, M., Surdeanu, M., Mihalcea, R., Gîrju, R., Rus, V., Lăcătusu, F., Morărescu, P., and Bunescu, R. C. (2001b). Answering complex, list and context questions with lcc's question-answering server. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*.

Harabagiu, S. M., Paşca, M. A., and Maiorano, S. J. (2000). Experiments with open-domain textual question answering. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 292–298.

Hardy, H., Shimizu, N., Strzalkowski, T., Ting, L., Zhang, X., and Wise, G. B. (2002). Cross-document summarization by concept classification. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128.

Hickl, A., Lehmann, J., Williams, J., and Harabagiu, S. (2004). Experiments with interactive question answering in complex scenarios. In *Workshop on Pragmatics of Question Answering, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 60–69.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y. (2000). Question answering in Webclopedia. In *Proceedings of the Ninth Text Retrieval Conference (TREC 2000)*.

Hovy, E., Hermjakob, U., and Lin, C.-Y. (2001). The use of external knowledge in factoid QA. In *Proceedings of the Tenth Text Retrieval Conference, National Institute of Standards and Technology (NIST), Nov. 13-16, Gaithersburg, Maryland, USA (TREC 2001)*, pages 644–652.

Huang, T.-M., Kecman, V., and Kopriva, I. (2006). *Kernel Based Algorithms for Mining Huge Data Sets*. Springer, Berlin, Germany.

Ittycheriah, A., Franz, M., and Roukos, S. (2001). IBM's statistical question answering system - TREC-10. In *Proceedings of the Tenth Text Retrieval Conference, National Institute of*

Standards and Technology (NIST), Nov. 13-16, Gaithersburg, Maryland, USA (TREC 2001), pages 258–264.

Jaakkola, T., Diekhans, M., and Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158.

Joachims, T. (2002). SVM^{light} support vector machine.

Klein, D., Smarr, J., Nguyen, H., and Manning, C. D. (2003). Named entity recognition with character-level models. In *Proceedings of 7th Conference on Computational Natural Language Learning*, pages 180–183.

Kohomban, U. S. and Lee, W. S. (2005). Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 34–41.

Lee, G. G., Seo, J., Lee, S., Jung, H., Cho, B.-H., Lee, C., Kwak, B.-K., Cha, J., Kim, D., An, J., Kim, H., and Kim, K. (2001). SiteQ: engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In *Proceedings of the Ninth Text Retrieval Conference (TREC 2000)*.

Lin, C.-Y. (1999). Training a selection function for extraction. In *Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM)*, pages 55–62.

Lin, C.-Y. (2004). ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Workshop on Text Summarization Branches Out*.

Lin, D. (1994). Principar – an efficient, broad-coverage, principle-based parser. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 482–488.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 768 – 774.

Lin, J. and Demner-Fushman, D. (2005). Evaluating summaries and answers: Two sides of the same coin? In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

Ma, J., Zhao, Y., Ahalt, S., and Eads, D. (2003). OSU SVM classifier Matlab toolbox. In *http://svm.sourceforge.net/docs/3.00/api/*.

Malioutov, I. and Barzilay, R. (2006). Minimum cut model for spoken lecture segmentation. In *Proceedings of 44th Annual Meeting of the Association for Computational Linguistics*.

Mann, W. and Thompson, S. (1986). *Rhetorical Structure Theory: Description and Construction of Text Structures*. Kluwer Academic Publishers, Boston, U.S.A.

Manning, C. and Sch´utze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, U.S.A.

Marcu, D. and Echihabi, A. (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375.

Mendonça, E. A., Cimino, J. J., Johnson, S. B., and Seol., Y.-H. (2001). Accessing heterogeneous sources of evidence to answer clinical questions. *Journal of Biomedical Informatics*, 34:85–98.

Moldovan, D. and Harabagiu, S. M. (2000). The structure and performance of an open-domain question answering system. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics, Oct. 1-8, 2000, Hong Kong, China (ACL-2000)*, pages 563–570.

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

MUC (1995). Message Understanding Conferences. In *http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html*.

National Library of Medicine (2004). SPECIALIST parser. In *SPECIALIST Text Tools*.

Niu, Y. and Hirst, G. (2004). Analysis of semantic classes in medical text for question answering. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Workshop on Question Answering in Restricted Domains*, pages 54–61.

Niu, Y., Hirst, G., McArthur, G., and Rodriguez-Gianolli, P. (2003). Answering clinical questions with role identification. In *Proceedings of 41st Annual Meeting of the Association for Computational Linguistics, Workshop on Natural Language Processing in Biomedicine*, pages 73–80.

Niu, Y., Zhu, X., and Hirst, G. (2006). Using outcome polarity in sentence extraction for medical question-answering. In *Proceedings of the American Medical Informatics Association 2006 Annual Symposium*, pages 599–603.

Niu, Y., Zhu, X., Li, J., and Hirst, G. (2005). Analysis of polarity information in medical text. In *Proceedings of the American Medical Informatics Association 2005 Annual Symposium*, pages 570–574.

Nomoto, T. and Matsumoto, Y. (2001). A new approach to unsupervised text summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 26–34.

Nomoto, T. and Matsumoto, Y. (2003). The diversity-based approach to open-domain text summarization. *Information Processing and Management: an International Journal*, 39(3):363–389.

Paşca, M. and Harabagiu, S. M. (2001a). The informative role of wordnet in open-domain question answering. In *Proceedings of the NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pages 138–143.

Paşca, M. A. and Harabagiu, S. M. (2001b). High performance question/answering. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA (SIGIR-2001)*, pages 366–374.

Pang, B. and Lee, L. (2004). A sentimental education: sentiment analysis using subjectivity smmarizaiton based on minimum cuts. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 271–278.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

Prager, J., Brown, E., and Coden, A. (2000). Question–answering by predictive annotation. In *Proceedings of the 23th International Conference on Research and Development in Information Retrieval (SIGIR-2000)*, pages 184–191.

Purver, M., Körding, K., Griffiths, T., and Tenebaum, J. (2006). Unsupervised topic modelling for multi-party spoken discourse. In *Proceedings of 44th Annual Meeting of the Association for Computational Linguistics*.

Ray, S. and Craven, M. (2001). Representing sentence structure in hidden Markov models for information extraction. In *Proceeding of 17th International Joint Conferences on Artificial Intelligence*, pages 1273–1279.

Richardson, W. S., Wilson, M. C., Nishikawa, J., and Hayward, R. S. (1995). The well-built clinical question: a key to evidence-based decisions. *ACP J Club*, 123(3):12–13.

Riloff, E. (1999). Information extraction as a stepping stone toward story understanding. *Computational Models of Reading and Understanding*.

Rosario, B. and Hearst, M. A. (2004). Classifying semantic relations in bioscience texts. In *Proceedings of 42nd Annual Meeting of the Association for Computational Linguistics*, pages 431–438.

Rosenberg, A. and Hirschberg, J. (2006). Story segmentation of broadcast news in english, mandarin, and arabic. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 125–128.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Sackett, D. L. and Straus, S. E. (1998). Finding and applying evidence during clinical rounds: the "Evidence Cart". *Journal of the American Medical Association*, 280(15):1336–1338.

Sackett, D. L., Straus, S. E., Richardson, W. S., Rosenberg, W., and Haynes, R. B. (2000). *Evidence-Based Medicine: How to Practice and Teach EBM*. Harcourt Publishers Limited, Edinburgh.

Saito, M., Yamamoto, K., and Sekine, S. (2006). Using phrasal patterns to identify discourse relations. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 133–136.

Sang, E. F. T. K. and Meulder, F. D. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of Conference on Computational Natural Language Learning*, pages 142–147.

Sekine, S. (1997). Apple pie parser. In *http://nlp.cs.nyu.edu/app/*.

Small, S., Strzalkowski, T., Liu, T., Ryan, S., Salkin, R., Shimizu, N., Kantor, P., Kelly, D., Rittman, R., Wacholder, N., , and Yamrom, B. (2004). Hitiqa: Scenario based question answering. In *Workshop on Pragmatics of Question Answering, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 52–59.

Soricut, R. and Brill, E. (2006). Automatic question answering using the web: Beyond the factoid. *Information Retrieval – Special Issue on Web Information Retrieval*, 9:191–206.

Soubbotin, M. M. (2001). Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the Tenth Text Retrieval Conference, National Institute of Standards and Technology (NIST), Nov. 13-16, Gaithersburg, Maryland, USA (TREC 2001)*, pages 293–302.

Srihari, R. and Li, W. (1999). Information extraction supported question answering. In *Proceedings of the Eighth Text Retrieval Conference, National Institute of Standards and Technology (NIST), Nov. 17-19, Gaithersburg, Maryland, USA (TREC 1999)*, pages 185–196.

Stoyanov, V., Cardie, C., and Wiebe, J. (2005). Multi-perspective question answering using the opqa corpus. In *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing*.

Straus, S. E. and Sackett, D. L. (1999). Bringing evidence to the point of care. *Journal of the American Medical Association*, 281:1171–1172.

TREC (2001). Text retrieval conference. In *http://trec.nist.gov/*.

Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Valin, V. and Robert, D. (1993). A synposis of role and reference grammar. In Robert, D. and Valin, V., editors, *Advances in Role and Reference Grammar*, pages 1 – 166. John Benjamins Publishing Company, Amsterdam.

Voorhees, E. M. (2001). Overview of the TREC 2001 question answering track. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631.

Yang, X., Su, J., and Tan, C. L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics*.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. In *Carnegie Mellon University, Center for Automated Learning and Discovery, Technical Report CMU-CALD-02-107*.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*.