Feature Engineering for Domain Independent Named Entity Recognition and Biomedical Text Mining Applications

György Szarvas

Research Group on Artificial Intelligence

of the Hungarian Academy of Sciences
and the University of Szeged

June 2008

A thesis submitted for the degree of doctor of philosophy
of the University of Szeged

University of Szeged
Doctoral School in Mathematics and Computer Science
Ph.D. Programme in Informatics

Abstract

Information searches in unstructered data formats like text documents are often performed by humans in many fields. Since such tasks are usually beyond the capability of keyword-lookup based processing, they are laborious and time consuming to do manually and thus computer-assisted or automatic solutions are desperately needed. Text Mining attempts to provide solutions for these tasks.
Typical Text Mining tasks are:

- Finding the names of relevant objects / entities in the text (Named Entity Recognition)

- Evaluating the document and carrying out a certain action based on its content (Text Categorisation / Document Classification)

These tasks can be formulated as classification tasks in Machine Learning terminology and this means that – when we have a set of pre-processed (labeled) examples on hand – they can be solved by statistical or rule-based systems that discover patterns and regularities in the labeled set of examples and exploit this knowledge to process new documents. This way human labour can be replaced or at least made more efficient.
In this thesis

- we address specific problems that fall into the above-mentioned categories.

- we design an appropriate feature space representation (via feature generation, selection, etc.) that permits the development of efficient statistical or rule-based solutions for these tasks.

- we evaluate our models (and thus the feature representations we developed) on standard evaluation datasets to demonstrate the usefulness of our systems.

We present from a feature representation point of view, several practical text mining applications developed together with colleagues. The applications themselves cover a wide range of tasks from entity recognition (word sequence labelling) to multi-label document classification and also cover different domains (from business news texts to medical records / biological scientific articles). Our aim was to demonstrate that task-specific feature engineering is beneficial to the overall performance and that for specific text mining tasks one can construct systems that are useful in practice and even compete with humans in processing textual data.

György Szarvas, June 2008.

Acknowledgements.

First of all, I would like to thank my supervisor, Prof. János Csirik, for his guidance and for supporting my work with useful comments and letting me work at an inspiring department, the Research Group on Artificial Intelligence.

I would also like to thank my colleagues and friends who helped me to realise the results presented here and to enjoy my PhD study at the University of Szeged. In alphabetical order: András Bánhalmi, Róbert Busa-Fekete, Richárd Farkas and Róbert Ormándi.

I am indebted to the organisers of the evaluation campaigns that produced the datasets which enabled me to work on the extremely challenging and interesting problems discussed here and my linguist collegues whose endeavours were indispensable in addressing new tasks and creating useful corpora for the research community.

I would also like to thank David P. Curley for scrutinizing and correcting this thesis from a linguistic point of view.

I would like to thank Kati for her endless love, support and inspiration. Last, but not least I wish to thank my parents, my grandparents and my brother for their constant love and support. I would like to dedicate this thesis to them as a way of expressing my gratitude and appreciation.

List of Figures

List of Tables

Contents

Motivation - feature engineering in machine learning systems

The goal of pattern recognition for data mining (DM) is to induce such models based on previously known examples which capture / express non-trivial knowledge about the objects observed that can be utilised in processing and analysing previously unknown examples.

The field of Machine Learning (ML) seeks to develop computational methods that simulate human learning in the sense of extracting information from data automatically, by computational and statistical methods. Data Mining makes extensive use of machine learning techniques to solve particular tasks. Specific data mining tasks are the so-called classification problems where the knowledge to be extracted can be formulated as a class label of the objects observed; i.e. the goal is to build models for the classification of unseen objects into a limited number of pre-defined categories based on a set of pre-classified examples. These classification models exploit characteristic attributes of the examples that are given or are measurable for new instances as well. In ML terminology these attributes are called features. Thus, solving a practical machine learning task can be divided into two major steps, namely feature engineering (when we define the features used for classification) and model construction (when we select a concrete learning algorithm, its parameters and then train a classifier to solve a given task).

Obviously, if we have very good features that are strongly related to the class labels that need to be induced, the classification task itself will be straightforward. The general goals of these steps are complementary in the sense that having an ideal feature representation of the problem results in a trivial classification task as the information relevant for classifying the observations is explicitly given in the feature representation and any (even a very simple) classifier algorithm is capable of utilising this knowledge.

On the other hand, the feature representation may be ill-suited to the problem in practice, i.e. the learning problem remains complex. The attributes we can define and measure for the observed objects may lack a piece of the knowledge that is important for solving the problem, and the feature values might contain noise (as measurements usually do). In such cases the selection of a proper classifier algorithm or combination of algorithms and system parameters becomes important if we wish to achieve a good overall performance.

Motivation - text mining applications

The amount of information stored in electronic document form is growing at an exponentially increasing rate. Searching for and processing information from textual sources is increasingly time-consuming in many areas (like medicine, research or business) and is becoming infeasible to perform without computer assistance. Thus today the need for intelligent text processing applications that can supersede or assist human information search in text documents is strong. Even though the performance of computers is not as good as the performance of humans in most complex information processing tasks, computers also have some obvious advantages to humans in their capacity of processing and the precision in performing well-defined tasks (e.g. indexing the whole World Wide Web).

An emerging field of Natural Language Processing is Text Mining, which seeks to automatically process large amounts of unstructured text; that is, to locate and classify relevant items of information and populate some sort of structured data collection for human processing. This task obviously requires at least a limited understanding of the text itself and the induction of new complex patterns that simulate human information search, which makes text mining tasks more complex and challenging than traditional keyword-lookup based information retrieval tasks.

Aim of this thesis

In this thesis we present several practical text mining applications developed together with colleagues, from a feature representation point of view. The applications themselves cover a wide range of different tasks from entity recognition (word sequence labelling) to multi-label document classification and different domains (from business news texts to medical records / biological scientific papers). Our aim is to demonstrate that task-specific feature engineering is beneficial to the overall performance and for specific text mining tasks, it is feasible to construct systems that are useful in practice and even compete human performance in processing the majority of textual data.

Structure of the thesis

This thesis is organized into seven main chapters. The first, introductory chapter briefly introduces the topics we discuss later on and summarises the main characteristics of the solution we gave for each problem addressed. At the end of the Introduction we discuss the most important contributions of the author to the research and development described in the subsequent chapters, and we also list the author's contributions for each cited paper (the papers that discuss the same topics and results as discussed in the thesis).

The remaining six chapters are grouped to two major thesis parts, one about our Named Entity Recognition (chapters 2-4) work and the other about our research in Text Classification (chapters 5-7).

The second chapter introduces the Hungarian Named Entity Corpus we developed and our Hungarian Named Entity Recognition approach along with the results we got in Hungarian NER. The feature set we implemented for NER is discussed in details here.

The third chapter describes our English Named Entity Recognitition model, the extensions of the feature set designed for Hungarian that proved to be neccessary to achieve a good performance. Our research in Named Entity lemmatisation (removal of inflections from proper names) for English (and also applied for Hungarian) is also discussed in this chapter.

The fourth chapter describes the domain adaptation and extension of our system for anonymisation of medical discharge summaries. The novel features developed for the medical domain and a dynamic learning/feature generation approach is introduced in this chapter.

The fifth chapter focuses on the classification of medical discharge summaries according to the patient's smoking status. Our novel features and feature selection experiments are discussed in detial in this chapter.

The sixth chapter focuses on the classification of sentences according to their modality (they contain a speculative part or not). Our novel feature selection method and weakly supervised / unsupervised training data generation method are discussed in detial in this chapter. We also introduce here the corpus we built for the detection of the linguistic scope pf speculative and negative cues.

The seventh chapter's topic is clinical coding of medical documents using hybrid (rule-based and statistical) models. The development of rule-based systems from online sources, our comination approach and our experiments in automatic discovery of label-dependencies are discussed in detail here.

The end of the thesis includes an Appendix chapter (where all the specific terms we used throughout the thesis are introduced) and a Summary of the whole study (both in English and Hungarian).

Chapter 1

Introduction

1.1 Entity Recognition tasks

The identification and classification of rigid designators [1] like proper nouns, acronyms of names, time expressions, measures, IDs, e-mail addresses and phone numbers in plain text is of key importance in numerous natural language processing applications. The special characteristic of these rigid designators (as opposed to common words) is that they have no *meaning* in the traditional sense but they refer to one or more entities of the world uniquely (references). These text elements are called Named Entities (NEs) in the literature.

Named Entities generally carry important information about the text itself, and thus are targets for Text Mining [2]. Another task where NEs have to be identified is Machine Translation which has to handle proper nouns and common words in a different way due to the specific translation rules that apply to names [3]. Entity co-reference [4] and disambiguation [5] (two related tasks) is also an important task in Information Retrieval since a major part of queries are entity names that are highly ambiguous.

Because of the above, the very first step in almost every Text Mining task is to detect names in the text that belong to task-specific entity types. These tasks are the so-called Named Entity Recognition (NER) problems, where one tries to recognise (single or subsequent) tokens in text that together constitute a rigid designator phrase, and to determine the category type to which these phrases belong. Categorisation is always task specific, as different kinds of entities are important in different domains. Sometimes entity recognition itself can be a standalone application, as in the case of anonymisation issues, where no further processing is required when all the name phrases have been located in the text.

NER tasks can be solved using labelled corpora and statistical methods that induce NE tagging rules by discovering patterns in the manually annotated source text. Being a simple but crucial task, NER has been evaluated for various domains and languages. The variety of languages for which a major evaluation campaign include English [6], German [7], Dutch, Spanish [8], Chinese [9], Japanese [10] just to name a few, while domains where NER has been (and is) studied extensively includes the task of processing economic, sports and political news [7], medical texts [11], chemical [12], biological texts [13] or military documents [14]. In this study we deal with Hungarian NER and English newswire and medical NER.

Though the nature of the information that is important and is thus the target for recognition differs from application to application, these tasks can be handled with such models that are, in a limited sense, language [7] [15] (and domain [16] [17]) independent. The language and domain independence of NER systems means that a similar algorithm is capable of solving various tasks, independently of the target language and domain, as long as a labelled corpus for a particular language/domain pair is available and the entity types to be recognised are more or less similar. Cross language and/or cross domain recognition where systems are trained and used in different languages or domains has also been widely studied, but such scenarios are beyond the scope of this work, hence they will not be discussed here. We will focus on the recognition of proper names in multiple languages (Hungarian and English) and multiple domains (newswire and medical texts) and the recognition of a few other entity types like dates, IDs, etc. in English medical documents.

1.1.1 Example name tagging tasks

Now we will give a very brief introduction to some specific name tagging tasks in order to give the reader a better insight into the nature of NER tasks. The examples listed here include those problems that we will address later on in this thesis.

English newswire NER

We dealt with the NER evaluation task of the Computational Natural Language Learning (CoNLL) 2003 [7] conference for English language. Here the goal was the correct identification of personal, organization and location names, along with other proper nouns treated as miscellaneous entities in texts of news press releases of Reuters Inc. from 1996.
An example of English NER is

- [U.N.]$_{ORGANISATION}$ official [Rolf Ekeus]$_{PERSON}$ heads for [Baghdad]$_{LOCATION}$.

Hungarian newswire NER

We addressed a NER task similar to the CoNLL 2003 guidelines for Hungarian language[18]. Thus we had to distinguish between person, organization and location names, and miscellaneous entities, and used texts from the Szeged TreeBank[19] of short press releases of the Magyar Távirati Iroda.
An example of Hungarian NER is

- A pénzügyi kockázatok kezeléséről kétnapos nemzetközi konferenciát tartanak csütörtökön és pénteken [Budapesten]$_{LOCATION}$ - mondta [Kondor Imre]$_{PERSON}$, a [Magyarországi Kockázatkezelők Egyesületének]$_{ORGANISATION}$ elnöke szerdán [Budapesten]$_{LOCATION}$ a sajtótájékoztatón.

English medical NER

For medical texts, an important use of NER is the automatic anonymisation of medical reports, to facilitate information exchange/access and respect individual patient rights (the protection of personal data). According to the guidelines of Health Information Portability and Accountability Act (HIPAA) of the US, the medical records released must be free of seventeen categories of textual Personal Health Information (PHI), out of which 8 actually appeared in the discharge summaries we used: first and last names of patients, their health proxies, and family members; the patient's age (if above 89 years old); doctors' first and last names; identification numbers; telephone, fax, and pager numbers; hospital names; geographic locations; and dates. To develop and test our model we used the de-identification dataset of the I2B2 Workshop on Challenges in Natural Language Processing for Clinical Data [11].
An example of the de-identification task is

- Mr. [Cornea]$_{PATIENT}$ underwent an ECHO and endoscopy at [Ingree and Ot of Weamanshy Medical Center]$_{HOSPITAL}$ on [April 28]$_{DATE}$.

1.1.2 The statistical NER system we developed

To solve the above problems we developed a statistical NER system that performed well across languages (English and Hungarian) and domains (newswire press releases and medical reports) with only slight modifications needed to port the system from the newswire domain to medical texts - we added a few features that exploit the specific characteristics of medical texts and a loop to the training process to achieve an even better performance. Our results showed that the model was successful even without these (fine tuning) domain extensions.

The NER system we developed treats the NER problem as the classification of separate tokens. Using labeled corpora of about 200000 tokens in size, we applied a decision tree classifier (C4.5 [20]) and boosting (AdaBoostM1 [21]) to NER, two algorithms that are well-known from the machine learning literature.
To solve a similar NER problem in different settings, we use the same learning model, and the same or very slightly modified feature set. Of course, most features that have an external source (lists, frequency information, etc.) are customized to the actual task by using a different source for calculating feature values, i.e. Hungarian NER uses Hungarian lists, English NER uses English lists, medical NER uses medical term lists, etc. Our general classifier model exploits features of several different types (a more detailed description is given in the corresponding thesis chapter), including:

- **gazetteers** of unambiguous NEs from the train data: we used the NE phrases which occur more than five times in the train texts and had the same label in over 90% of the cases,

- **dictionaries** of first names, company types, sport teams, denominators of locations (mountains, city) and so on: we collected special English lists from the Internet,

- **orthographical features**: capitalization, word length, common bit information about the word form (contains a digit or not, has an uppercase character inside the word, regular expressions and so on). We collected the most representative character level bi/trigrams from the train texts assigned to each NE class,
- **frequency information**: frequency of the token, the ratio of the token's capitalized and lowercase occurrences, the ratio of capitalized and sentence start frequencies of the token,
- **phrasal information**: chunk codes and forecasted class of a few preceding words (we carried out an online evaluation),
- **contextual information**: POS codes, sentence position, document zone (title or body), topic code, trigger words (the most frequent and unambiguous tokens in a window around the NEs) from the train text and whether the word is inside quote marks or not.

Owing to the beneficial characteristics of decision tree learning and the compact feature representation we developed (using fewer than 200 features for the general NER task, and omitting the word form itself from the classification process), our model is fast to train and evaluate, and performed well on standard evaluation datasets.
Our domain and language independent model achieved:

- an 89.02% F measure on the CoNLL 2003 evaluation set
- a 94.76% F measure on the Hungarian Named Entity Corpus
- a 94.34% F measure on the de-identification challenge of the I2B2 workshop.

Domain extensions improved the performance of our system on medical texts and our model gave an F measure of 97.64%. All these evaluations correspond to phrase-level equal-weighted F measures on each entity class.

1.2 Assertion/Document classification tasks in biomedical texts

The human processing of textual data (system logs, medical reports, newswire articles, customer feedback records, etc.) is a laborious and costly process, and is becoming unfeasible with the increasing amount of information stored in documents. There is a growing need for solutions that automate or facilitate the information processing workflow that is currently performed by humans. Thus today the automatic classification of free texts (either assertions or longer documents) based on their content and converting textual data to practical knowledge is an important subtask of Information Extraction.

Many text processing tasks can be formulated as a classification problem and solved effectively with Machine Learning methods [22] that are capable of uncovering the hidden structure in free text, assuming that labelled examples are on hand to train the automatic systems on. These solutions go one step beyond simple information retrieval (that is, providing the user with the appropriate documents using keyword lookup and relevance ranking), as they require the (deep or shallow) understanding of the text itself. The systems have to handle synonymy, transliterations and language phenomena like negation, sentiment, subjectivity and temporality [23].

A major application domain of practical language technology solutions is the field of Biology and Medicine [24]. Experts in these fields usually have to work with large collections of documents in everyday work in order to carry out efficient research (reading scientific papers, patents, or reports on earlier experiments in the subject) or decision making (reports on examination of former patients with similar symptoms or diseases).

Even though language or domain independent models would be desirable as well, solutions of such generality are in many cases beyond the scope of current state-of-the-art NLP technology. Economic aspects thus motivate the development of more specific solutions for unique concrete problems. In this thesis we will focus on the issues associated with biological and medical text processing.

1.2.1 Example text classification tasks

Here we will give a brief introduction to some specific tasks in order to give the reader a better insight into the nature of assertion or document level text classification. The examples listed here include those problems that will be addressed later on in this thesis.

Smoker status extraction from medical discharge summaries

The main purpose of processing medical discharge records is to facilitate medical research carried out by physicians by providing them with statistically relevant data for analysis. An example of such an analysis might be a comparison of the runoff and effects of certain diseases among patients with different social habits [25]. The evidence drawn from the direct connection between social characteristics and diseases (like the link between smoking status and lung cancer or asthma) is of key importance in treatment and prevention issues.

Such points can be deduced automatically by applying statistical methods on large corpora of medical records. Here we used the 'smoker status' dataset of the I2B2 Workshop on Challenges in Natural Language Processing for Clinical Data [26]. The task in this case is to classify the medical records into the following five semantic classes based on the smoking status of the patient being examined:

- non-smoker: the patient has no smoking history,
- current smoker: he/she is an active smoker,
- past smoker: the patient had not smoked for at least one year,
- smoker: when the document contains no information about his current or past smoker status, but he/she has smoking history,

- unknown: the report contains no information about the patient smoking status.

A sample assertion on patient smoking status in a discharge summary is:

- *The patient is a 60 yo right handed gentleman with a 20-years history of heavy smoking. Agreed to participate in a smoking cessation program. (current smoker)*

Detection of speculations in assertions

The highly accurate identification of several regularly occurring language phenomena like the speculative use of language [27] [28], negation and past tense (temporal resolution) is a prerequisite for the efficient processing of biomedical texts. In various Text Mining tasks, relevant statements appearing in a speculative context are treated as false positives. Hence hedge detection seeks to perform a kind of semantic filtering of texts; that is it tries to separate factual statements from speculative/uncertain ones. For biological scientific texts, we used a corpus consisting of articles on the fruit fly, provided by [29], and also used a small annotated corpus of 4 BMC Bioinformatics articles for external-source-evaluation. To evaluate our models in the medical domain, we used the standard dataset provided for the International Challenge on Classifying Clinical Free Text Using Natural Language Processing and a rule-based ICD-9 coder system constructed by us to provide false positive ICD-9 labels for automatic hedge dataset generation.
Two examples of speculative assertions in biological scientific texts are:

- *Thus, the D-mib wing phenotype may result from defective N inductive signaling at the D-V boundary.*

- *A similar role of Croquemort has not yet been tested, but seems likely since the crq mutant used in this study (crqKG01679) is lethal in pupae.*

Two examples of speculative assertions radiology reports are:

- *Findings suggesting viral or reactive airway disease with right lower lobe atelectasis or pneumonia.*

- *Right middle lobe infiltrate and/or atelectasis.*

Automatic ICD-9-CM coding of radiology reports

The assignment of International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM) codes serves as a justification for carrying out a certain procedure. This means that the reimbursement process by insurance companies is based on the labels that are assigned to each report after the patient's clinical treatment. The approximate cost of ICD-9-CM coding clinical records and correcting related errors is estimated to be about $25 billion per year in the US [30].

Since the ICD-9-CM codes are mainly used for billing purposes, the task itself is commercially relevant: false negatives (i.e. overlooked codes that should have been

coded) will cause a loss of revenue to the health institute, while false positives (the over coding of documents) is penalised by a sum three times higher than that earned with the superfluous code, and also entails the risk of prosecution to the health institute for fraud. Thus there is a desperate need for high-performance automatic ICD-9 coding systems. Here we used the standard dataset provided for the shared task International Challenge on Classifying Clinical Free Text Using Natural Language Processing on ICD-9-CM coding of radiology reports [31].
Two examples of ICD-9-CM coding of radiology reports are:

- *CODES: 486; 511.9*
  HISTORY: Right lower lobe pneumonia, cough, followup.
  IMPRESSION: Persistent right lower lobe opacity with pleural effusion present, slightly improved since prior radiograph of two days previous.

- *CODES: 593.89; V13.02*
  HISTORY: 14-year - old male with history of a single afebrile urinary tract infection in January with gross hematuria for a week. The patient was treated with antibiotics.
  IMPRESSION: Mild left pyelectasis and ureterectasis. Otherwise normal renal ultrasound. The bladder appears normal although there is a small to moderate post void residual.

1.2.2 The statistical assertion/document classifier systems we developed

To solve the above tasks and get a satisfactory performance we developed machine learning models for each that differ slightly from each other. The cost of building corpora of reasonable size for these tasks is very high; and thus to have labeled corpus of a similar size to that for entity tagging problems was not deemed feasible. Instead, we had to develop solutions that can infer the structure and knowledge hidden in the text from a few hundred (or few thousand) examples. Another option is to gather labeled examples fully or semi-automatically (within the scope of weakly supervised learning), but in such cases significant noise in the semi-automatic labeling has to be dealt with. Despite the nice results we obtained on these tasks the portability of the learned hypotheses suffered from this lack of training data, as our experiments revealed.

Smoker status extraction from medical discharge summaries

To classify smoker status in discharge summaries, we applied a sentence-level classifier based on the VSM representation of discharge summaries. We extended a unigram representation with complex features of pre-classified bigrams and trigrams (based on their meaning) and simple syntactic features plus negation detection as well. Our experiments showed that the features we introduced for smoker status classification were more helpful for learning than a simple VSM used by earlier approaches (feature selection methods chose our complex features more often than unigrams). The final classification of our system was based on the majority voting of several classifiers (C4.5 decision tree, Multi-Layer Perceptron and Support Vector Classifier). The solution we proposed for smoker status detection proved to be particularly efficient in discarding irrelevant documents (unknown class) and non-smokers, thus it could be used as a pre-processor for human processing/annotation. Our model achieved an overall accuracy of 86.54% on the I2B2 challenge evaluation set, close to the best solution that was entered in the challenge.

Detection of speculations in assertions

Here we used weakly supervised settings for biological texts and no supervision for medical data to acquire training sets for detecting hedges in biological or medical texts (at the sentence level). To classify sentences into speculative and non-speculative assertions we applied a Maximum Entropy classifier [32] and vector space model (VSM) to represent the examples. Uni-, bi- and trigrams of words were all included in the VSM representation. After a careful selection of relevant hedge keywords that involved a ranking and filtering of keywords via a modified class-conditional probability score (only the best 2 keywords received credit for appearing in speculative sentences) and then sorting the best candidates according to the $P(spec)$ scores given by the Maximum Entropy classifier, we managed to filter out the best speculative keywords from a training set constructed by using minimal supervision (biological scientific texts) or no supervision at all (radiology reports). This procedure resulted in an $F_{\beta=1}(spec)$ score of 85.08% for biological scientific papers, and an $F_{\beta=1}(spec)$ score of 82.07% for clinical free-texts.

Having selected the most relevant keywords, our hedge classifier simplified to a simple keyword matching routine that predicted speculative label every time a strong keyword was present. We suggest the use of 2 or 3 word long phrases to capture the rare non-speculative uses of otherwise strong keywords as a possible solution for further improving the model. This idea would require a labeled corpora of reasonable size to implement and evaluate, though.

The automatic ICD-9-CM coding of radiology reports

For the automated clinical coding of medical free texts we applied a hybrid rule-based and statistical model. A unique feature of the task was that expert coding guides that describe the principles of ICD-9-CM coding for humans were on hand and these guides were good sources for the swift implementation of an ICD-9-CM coding expert system. Thus in our study we focused on the possible ways of exploiting labeled data to fine-tune such an expert rule-based system.

First we developed a simple rule-based system for ICD-9 coding using one of the several online coding guides available. Then we trained statistical models using the codes assigned by the rule-based system to model the inter-label dependencies between disease and associated symptom codes. To address the second major deficiency of rule-based systems based on coding guides – i.e. terminology missing from the lists in the coding guide (rare transliterations and abbreviations, etc.) – we also applied a statistical approach. In this case we trained classifiers for the false negative codes of the initial rule-based system.

With this hybrid rule-based and statistical model we got a very good performance, one very close to an entirely manually constructed system with a moderate development cost that would make the implementation of our system feasible for a large set of codes as well (while developing a hand-crafted system for several hundreds or thousands of codes would be problemmatic indeed).

1.3 Summary by chapters

Here we summarise our findings for each chapter of the thesis and provide the relation of each paper referred to in the thesis and the results described in different chapters in a table.

The thesis is divided into two major parts, one dealing with Named Entity Recognition problems and another that focuses on Text Classification tasks. Here we list our most important findings for each chapter.

- NER Chapters

    1. Hungarian NER Chapter

       For NER in Hungarian the author participated in the creation of the first Hungarian NER reference corpus which allowed researchers to investigate statistical approaches to Entity Recognition in Hungarian texts. This is a joint, inseparable contribution between the authors of [18] and the linguist colleagues who carried out the annotation work of the corpus.

       Together with his colleagues, the author designed a suitable feature representation for training machine learning models and set up an efficient learning model on the corpus that achieved a phrase level F measure performance of 94.76%. In the construction of the Named Entity Recognition system, the author made major contributions in designing the feature representation for learning algorithms.

       These results are described in [18], [33] and [34].

    2. English NER Chapter

       The author participated in adapting a NER system designed for the Hungarian language to a similar task in English. Together with his colleagues, the author extended the feature representation for training machine learning models and used the same, efficient learning model that was introduced for Hungarian NER. This system attained a phrase level F measure score of 89.02%.

       The author also participated in the development of a MaxEnt-based system for the metonymy resolution shared task of SemEval-2007 [35]. In the NE-metonymy classifier that was submitted to the challenge by the author and his colleagues, the author is responsible for the web-based approach that was designed to remove inflectional affixes from Named Entities and was used successfully as a feature to classify org-for-product metonymies.

When constructing the English Named Entity Recognition system, the author made major contributions in designing the feature extensions for learning algorithms.

The author investigated corpus frequency based heuristics that were capable of fine tuning NER systems by eliminating certain typical errors of NER systems. These heuristics were then altered to provide a heuristic solution to Named Entity lemmatisation, a problem that arises both in English (plural and possessive markers) and in Hungarian (agglutinative characteristic of the language). The author and his collegues showed that corpus statistics can be utilised to solve NE lemmatisation with good accuracy. The author's contribution is the idea and general concept of using web frequency counts for Named Entity lemmatisation (NE normalisation or affixes as features for other tasks).

These results are described in [34], [36], [37] and partly in [38].

3. Anonymisation of Medical Records

   Together with his collegues, the author participated in the 2006 I2B2 shared task challenge on medical record de-identification. The major parts of the adaptation of the pre-existing NER system, and the results achieved as a whole are the joint contribution of the co-authors. As our results show, the system we built via the domain adaptation of our newswire NER model is competitive with other approaches, which means that our architecture is capable of solving NER tasks language and domain independently, with minimal adaptation effort.

   In particular, the author made major contributions to the customisation of the feature representation, i.e. the development of novel features specifically for the medical domain. These novel features were helpful in achieving a state-of-the-art performance (our model had the best phrase-level 8-way F measure and second best token-level 9-way accuracy).

   These results are described in [39].

- Text Classification Chapters

  1. Smoker status classification in discharge summaries

     Together with his collegues, the author participated in the 2006 I2B2 shared task challenge on patient smoking status classification from medical records. The system and the overall results we submitted are a shared and indivisible contribution of the co-authors.

     In particular, the author made major contributions to the design of the feature representation, i.e. the development of features used by previous studies and novel ones specifically for the medical domain which tried to group more or less similar examples together by exploiting the syntactic or semantic classification of phrases. The main reasoning for having these novel features was to reduce the effects of a small sample size. These novel features were helpful in achieving a good performance (they appeared among the top ranked attributes using 2 different feature selection methods).

These results are described in [40].

2. Hedge Classification in biomedical texts

   All the contributions in the corresponding chapter are independent results of the author. The major findings of this thesis are the construction of a complex feature ranking and selection procedure that successfully reduces the number of keyword candidates (those having the highest class-conditional probability for hedge class) without excluding helpful hedge keywords.

   We also demonstrated that with a very limited amount of expert supervision in finalising the feature representation, it is possible to build accurate hedge classifiers from semi-automatically or automatically collected training data.

   We extended the scope of evaluations to two applications with different kinds of texts involved (scientific articles used in previous works, and also medical free texts).

   We extended the feature representation used by previous approaches to 2-3 word-long phrases and an evaluation of the importance of longer keywords in hedge classification.

   We demonstrated (using a small test corpora of biomedical scientific papers from a different source) that hedge keywords are highly task-specific and thus constructing models that generalise well from one task to another is not feasible without a noticeable loss in accuracy.

   These results are described in [41] and partly in [42].

3. ICD-9-CM coding in radiology reports

   Together with his collegues, the author participated in the 2007 CMC shared task challenge on automated ICD-9-CM coding of medical free texts using Natural Language Processing. The major steps of the development of the system as a whole that was submitted to the challenge, and the results achieved are a shared and indivisible contribution of the co-authors.

   In particular, the author made a major contribution to the development of a basic and an entirely hand-crafted rule-based classifier; the design, implementation and interpretation of the complex inter-annotator agreement analysis and the design of the machine learning model for discovering inter-label dependencies from the labeled corpus.

   These results are described in [43].

	HunNER	EngNER	DE-ID	SMOKER	HEDGE	ICD-9
LREC[18]	•					
ACTA[33]	•					
DS2006[34]	•	•				
SEMEVAL[38]		•				
ICDM2007[36]		•				
TSD2008[37]		•				
JAMIA[39]			•			
WSEAS[40]				•		
ACL[41]					•	
BIONLP[42]					•	
LBM2007[43]						•

Table 1.1: The relation between the thesis topics and the corresponding publications.

1.4 Summary by papers

Here we list the most important results in each paper that are regarded as the author's own contributions. We mention here that system performance scores (i.e. the overall results) are always counted as a shared contribution and not listed here, as several authors participated in the development of the systems described in the cited papers. The only exception is [41], which describes only the author's own results. [18] has been omitted from the list as all the results described in this paper are counted as shared contributions of the authors. For [38] the author made only marginal contributions.

- ACTA[33]
  - The construction of a feature representation for Hungarian NER.
  - Compact representation.
  - Frequency features.
- DS2006[34]
  - Description of feature space extensions for English NER.
- SEMEVAL[38]
  - The plural feature for NE-metonymy resolution.
- ICDM2007[36]
  - Using web frequency data for identifying consecutive NEs.
- TSD2008[37]
  - The idea and general concept of using web frequency counts for Named Entity lemmatisation (NE normalisation or affixes as features for other tasks)

- JAMIA[39]

  – The extension of the feature space with respect to the chief characteristics of medical texts.

  – The iterative learning/feature generation approach.

- WSEAS[40]

  – The use of token bi- and trigram features.

  – The use of deep knowledge features (pre-classified bigrams, syntactic information, negation).

  – The execution of feature selection methods for getting a suitable set of features for smoker classification.

- ACL[41]

  – All of the results in the paper.

- BIONLP[42]

  – Some of the general principles of negation, hedging and their scope annotation.

- LBM2007[43]

  – Detailed performance and annotator agreement analysis.

  – Complex features for discovering label-dependecies with machine leraning models.

  – The construction of a basic rule-based system that served as the basis for further developments, and an entirely hand-crafted system for comparison.

Part I

Named Entity Recognition

Chapter 2

Hungarian Named Entity Recognition

In this section we will introduce the Hungarian NER corpus, describe in detail the designed feature set for Hungarian NER. The feature set designed is regarded as one of the contributions of this thesis. We will analyze thoroughly the performance we got using the above feature set. At the end of this section we discuss the background of our results and draw conclusions on the potentials of machine learning approaches to NER problems in Hungarian.

2.1 The corpus

The Named Entity Corpus for Hungarian is a sub corpus of the Szeged Treebank [19][1], which contains 1.2 million words with tokenisation and full morphological and syntactic annotation done manually by linguist experts. A significant part of these texts was annotated with Named Entity class labels based on the annotation standards used on CoNLL conferences[2]. The corpus is available free of charge for research purposes.

2.1.1 General properties

Short business news articles collected from MTI (Hungarian News Agency, www.mti.hu) constitute a part of the Szeged Treebank, 225.963 words in size, covering 38 topics concerning the NewsML topic coding standard, ranging from acquisition to stock market changes or to new plant openings.

In the text we included annotations of person, location, organization names and miscellaneous entities that are proper names but do not belong to the three other classes. Part of speech codes generated automatically by a POS tagger [44] developed

[1] The project was carried out together with MorphoLogic Ltd. and the Hungarian Academy's Research Institute for Linguistics

[2] A brief description of the English data can be found in the next chapter of the thesis. A more detailed description of it and the data can also be accessed at http://www.cnts.ua.ac.be/conll2003/ner/

at the University of Szeged were also added to the database. In addition we provide some gazetteer resources in Hungarian (Hungarian first names, company types, list of names of countries, cities, geographical name types and a stopword list) that we used for experiments to build a model based on the corpus.

The dataset has some interesting aspects relating to the distribution of class labels which is induced by the domain specificity of the texts - organization class, which turned to be harder to recognize than person names for example, has higher frequency in this corpus than in other standard corpora for other languages.

2.1.2 Corpus example

An example sentence from the corpus (subsequent tokens with the same entity class label denote a single, longer entity phrase):
A 0
pénzügyi 0
kockázatok 0
kezeléséről 0
kétnapos 0
nemzetközi 0
konferenciát 0
tartanak 0
csütörtökön 0
és 0
pénteken 0
Budapesten I-LOC
- 0
mondta 0
Kondor I-PER
Imre I-PER
, 0
a 0
Magyarországi I-ORG
Kockázatkezelők I-ORG
Egyesületének I-ORG
elnöke 0
szerdán 0
Budapesten I-LOC
a 0
sajtótájékoztatón 0
. 0

We divided the corpus into 3 parts, namely a training, a development phase test and an evaluation subcorpus, following the protocol of the CoNLL-2003 NER shared

	Tokens	Phrases
non-tagged tokens	200067	–
person names	1921	982
organizations	20433	10513
locations	1501	1294
miscellaneous proper names	2041	1662

Table 2.1: Corpus details.

task.
Some simple statistics of the whole corpus and the three sub-corpora are:

	Sentences	Tokens
Training set	8172	192439
Development set	502	11382
Test set	900	22142

Table 2.2: The number of tokens and sentences in the corpus.

	LOC	MISC	ORG	PER
Training set	1148	1402	9212	886
Development set	33	138	373	19
Test set	113	122	928	77

Table 2.3: Number of named entities per data file.

2.1.3 The annotation process

As annotation errors can readily mislead learning methods, accuracy is a critical measure of the usefulness of language resources containing labelled data that can be used to train and test supervised Machine Learning models for Natural Language Processing tasks. With this we sought to create a corpus with as low an annotation error rate as possible, which could be efficiently used for training NE recognizer and classifier systems for Hungarian. To guarantee the precision of tagging we set up an annotation procedure with three stages.

In the first stage two linguists, who received the same instructions, labeled the corpus with NE tags. Both of them were told to use the Internet or other sources of knowledge whenever they were confused about their decision. Thanks to this and the special characteristics of the texts (domain specificity helps experts to become more familiar with the style and characteristics of business news articles); the resulting annotation was near perfect, in terms of inter-annotator agreement rate. We used the evaluation script made for the CoNLL conference shared tasks, which measures a phrase-level accuracy of a Named Entity-tagged corpus. The corpus showed an inter-annotator agreement of 99.6% after the first phase.

In the second phase all words that received different class labels were collected for discussion and revision by the two annotators and the chief annotator with several years of experience in corpus annotation. The chief annotator prepared the annotation guide and gave instructions to the other two to perform the first phase of labelling. Those entities that the linguists could not agree on initially received their class labels according to the joint decision of the group.

In the third phase all NEs that showed some kind of similarity to those that had been tagged ambiguously earlier were collected from the corpus for revision even though they received the same labels in the first phase. For example, if the tagging of shopping malls was inconsistent in a few cases (one annotator tagged *Árkád*$_{ORG}$ *bevásárlóközpont* while the other tagged *Árkád*$_{ORG}$ *bevásárlóközpont*$_{ORG}$), we checked the annotation of every appearance of every shopping mall name, regardless that the actual appearance caused disagreement or not. We did this to ensure the consistency of the annotation procedure. The resulting corpus after the final, third stage of consistency checking is considered error-free.

Creating error free resources of a reasonable size has a very high cost and, in addition, publicly available NE tagged corpuses contain some annotation errors, so we can say the corpus we developed has a great value for the research community of Natural Language Processing. As far as we know this is the only Hungarian NE corpus currently available, and its size is comparable to those that have been made for other languages.

2.1.4 Availability

The corpus is available for download and use for research or education purposes free of charge from the website of the Human Language Technology Group at the University of Szeged, Department of Informatics[3].

2.2 Description of our model and results

We regarded the NER problem essentially as the classification of separate tokens. We believe that this approach is competitive with the - theoretically more suitable - sequence labeling algorithms (like Hidden Markov Models, Conditional Random Fields); and we applied a decision tree learning algorithm. Of course our model is capable of taking into account the relationship between consecutive words using a window of appropriate size. We used 4 as a default window size for each feature.

Figure 2.1 shows the structure of our complex model; the details of *Feature generation* building block are described in this section, which is regarded as a contribution

[3] http://www.inf.u-szeged.hu/projectdirs/hlt/en/nercorpus.html

of the thesis, while the design of the learning model and classifier combination scheme is the contribution of the co-authors.

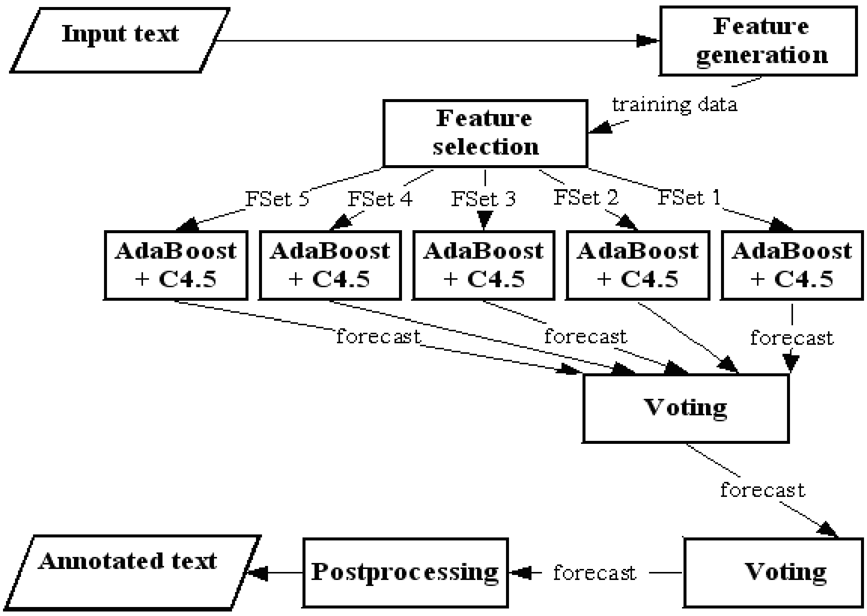


Figure 2.1: Outline of the structure of our NER model.

2.2.1 Boosting and decision trees for NER

Boosting (Shapire, 1990) and C4.5 (Quinlan, 1993) are well known algorithms for those who are acquainted with pattern recognition. Boosting has been applied successfully to improve the performance of decision trees in several NLP tasks. A system that made use of AdaBoost and fixed depth decision trees came first in the Computational Natural Language Learning Conference shared task on NER in 2002 (Carreras et al., 2002), but gave somewhat worse results in 2003 (it was ranked fifth with an F measure of 85.0% (Carreras et al., 2003)). We have not found any other competitive results for NER using decision tree classifiers and AdaBoost.

In our experiments 30 iterations of Boosting were performed on C4.5 decision trees (5 or more instances per leaf, pruning with confidence factor of 0.35 and subtree raising) as further iterations gave only a slight improvement.

2.2.2 Description of the feature space for NER

Here we will describe the feature set we designed for Hungarian NER and discuss its unique characteristics compared to representations introduced earlier. We will also analyse the typical performance that can be obtained using our representation, with a detailed comparison of features by relevance, novelty and performance gain.

Features used are:

- gazetteers and dictionaries: Gazeteers (lists of unambiguous entity names) and Dictionaries (lists of words that might benefit the recognition of NEs) are important and widely used NER features. We used the following gazetteers and lists:
  - gazetteers of unambiguous NEs from the train data: we used the NE phrases which occur more than five times in the train texts and got the same label in over 90% of the cases,
  - gazetteers of locations (Hungarian cities, world's largest cities, countries),
  - a dictionary of first names for Hungarian, English, French, German and Spanish
  - a dictionary of company types for several languages (e.g. *Kft* or *Zrt* for Hungarian and *Ltd*, *Corp*, ... for English)
  - a dictionary of geographical place name denominators (like *mountain*, *city*, *street*, etc.)
  - a dictionary of stopwords that can appear inside named (like *és*, *and*, *von*, etc.)
- orthographical features: Surface information like capitalisation carry much information about named entities. For example, capitalised words in the middle a sentence are usually entities. 3-4 letter-long, all-uppercase abbraviations usually denote organization or miscellaneous entities (brands), while 2 letter-long uppercase words are frequent abbreviations of locations (US states for example) or person names (the starting letters of first and family names). These observations encouraged us to describe the surface characteristics of words as detailed as possible. The ortographical features we used were the following:
  - capitalisation (is the first letter of the word capitalised or not)
  - word length (length of word in letters)
  - common bit information about the word form:
    - contains a digit or not
    - has an uppercase character inside the word
  - typical character bi/trigrams from the train texts assigned to each NE class
    - typical 2-4 letter-long suffixes (e.g. *vic* for Slavic or *son/sen* for northern person names)
    - character bi- and trigrams and 4-grams inside NE tokens (e.g. *ötv* for *kötvény/bond* appearing in miscellaneous and organization names)

- frequency information: frequency of the token, the ratio of the token's capitalised and lowercase occurrences, the ratio of capitalised and sentence beginning frequencies of the token, collected from the frequency data of the Szószablya korpusz [4]

- phrasal information: forecasted class labels of 4 preceding words (we performed an online evaluation),

- contextual information: POS codes (we used codes generated by our POS tagger for Hungarian instead of the existing tags from the Szeged Treebank), sentence position, trigger words (the most frequent and unambiguous tokens in a window around the NEs) from the train text, whether the word is between quotes or not.

Novelties in our feature representation

Probably the success of our feature representation lies in the fact that we utilised many different types of features in a compact representation. On the other hand, we also used the majority of the various features described earlier for NER, and also introduced some new ones (the frequency features) that actually turned out to be useful.

Compact representation

The compact representation means that we attempted to avoid using single tokens as standalone features. This means that we did not add the actual and surrounding word forms as binary features to describe the local context of the instance to be classified. Doing so would have introduced thousands of features even if we had constrained ourselves to using just the top ranked context-tokens (top ranked from the viewpoint of some measure of importance, e.g. the well-known TF-IDF measure). We also excluded single triggers or character level features from the model to avoid the blow-up in feature set dimensionality. Instead, we grouped together similar features (e.g. all character trigrams) that were semantically similar (they predicted the same class label/s/).
This way, instead of having several standalone features like:

- "The next token is Ltd" (indicative of ORG class)

- "The next token is Kft" (indicative of ORG class)

- "The next token is Corp" (indicative of ORG class)

- "The next token is City" (indicative of LOC class)
  and

- "The next token is Prix" (indicative of MISC class),

[4] http://mokk.bme.hu/projektek/szoszablya

we had a single feature (with the same number of possible values as the classes we differentiated):

- "The next token suggests that the actual token is a Named Entity of Class X"

This contraction of similar features which imply the same class label has two, contradictory effects. First, these features then become more informative as their coverage will be the sum of instances covered by any of the atomic features that were fused into one single attribute value. This definitely helps learning algorithms to incorporate knowledge from these attributes to the learned model. On the other hand, the contraction can have detrimental effects if a few of the atomic features that were integrated into the complex attribute become unstable. The noise in this case burdens the complex feature as a whole since atomic features become indistinguishable in the representation.

A third and probably most important consequence of this approach is that the dimensionality of the feature space representation is reduced considerably, which makes the training and testing of machine learning models faster (possibly at the cost of some loss in accuracy). The speed-up that can be achieved this way makes this approach favourable in many real-life scenarios where processing time is an important factor.

Frequency features

The frequency information of tokens collected from a very large corpus (large enough in size to represent the use of language in general) can hold much useful information for NER. This is especially true if the frequency of the same token is counted with both lowercase initial, uppercase initial occurences (the latter can be further identified as uppercase in the beginning or inside a sentence). For Hungarian NER, such information is available from the frequency dictionaries created in the Szószablya project.
We used frequency information in three different ways. These were:

- a **frequency count**: the number of occurences of the token (converted to lowercase). Very high values almost always indicate a non-entity token as these are typically the most extensively used common words of the language. On the other hand, low values often mean that the token is a (rare, perhaps foreign) named entity.

- ratios of **capitalised/lowercase** and **insentence-capitalised/all-capitalised** occurences: these two features together separate entity names very well from non-entities (without any specific use in further classification of entities) as NEs in Hungarian are the words that tend to occur in capitalised form (also in a non-sentence-starting position).

Silva et al. [45] used a feature similar to our capitalised/lowercase feature: they calculated the ratio of the frequency of the current token (e.g. *John*) and the case-insensitive frequency of the token (e.g. *john, JOHN, ...*). However, we used a huge text collection for calculating frequencies not the corpus used for NER, and we also distinguished sentence-beginning and inside-sentence appearances.

These features actually proved quite useful to our representation as the latter two ratio-features came among the top ranked features of all (in a χ^2 ranking).

2.3 Results

The results that are counted as the author's own contribution are presented above. Here we summarise the results of our system as a whole, which is a shared contribution of the co-authors of [33], [34]. Using the exact-match evaluation criterion of the CoNLL conference NER shared tasks, our model achieved an $F_{\beta=1} = 94.76\%$. These results are remarkably high, considering that the best results for NER in other languages using this evaluation measure are usually in the high 80s or low 90s. We should note here that our corpus is easier than many others for different languages due to the domain specificity of the corpus. The 11.38% higher performance of the same baseline system for Hungarian than for English supports this belief as well.

The per-class breakdown of $F_{\beta=1}$ scores are shown in the following table:

	Hungarian
LOC	95.07
MISC	85.96
ORG	95.84
PER	94.67
overall	94.76

Table 2.4: Per-class F scores.

2.4 Comparison and conclusions

Previous research on Hungarian name tagging includes expert rule-based approaches mainly because no labelled corpora of suitable size was available for training statistical models. To the best of the author's knowledge, the first name tagger was developed by researchers of the Hungarian Academy of Sciences at the Institute for Linguistics [46]. The system exploits regular patterns that capture named entity phrases in Hungarian texts.

Later on the Named Entity recogniser module of the HumorEsk [47] expert-rule based Hungarian syntactic parser was built based on Gábor et al.'s system.

A direct comparison between these systems and the statistical NER tagger described here is not easy to do as these systems were developed for general use and did not use the Hungarian NE corpus for the development of rule patterns. Undoubtedly, the system introduced here would perform better than the above-mentioned rule-based approaches on similar, short business news texts, while it would face major difficulties in labelling texts with really diverse characteristics – at least without any fine tuning (or retraining on another corpus).

The only other statistical NER system that was trained and tested on the same corpus using similar evaluation metrics is the system of Varga and Simon [48]. They trained their model using the same train/development/test splits of the same corpus, and trained a Maximum Entropy classifier based on their own feature representation. A fair comparison of the two systems is possible, their classifier yielding a 95.06% F measure on the same test set, which is slightly higher than the performance of our model. The difference in performance scores can be attributed to the different features and learning method they used.

Our findings above support the general view that – since named entities usually follow special orthographic and surface patterns – statistical approaches can handle name tagging tasks well and achieve a very good performance, even without the extensive use of Named Entity dictionaries (gazetteers). This observation has been reported for several languages and also holds for Hungarian NER. While providing a remarkably high performance, these systems have an advantage when they face rare entity names, where dictionary-based models can easily fail; and, using extensive local contextual patterns (previous labels, trigger words, etc.) makes statistical models more reliable on ambiguous entity names. On the other hand, traditional, dictionary-based NER taggers cannot handle ambiguous names that may fall into any of several different classes, depending on the context[5].

The results presented here are comparable with other state-of-the-art NER taggers and our system has some different characteristics from other statistical approaches for NER. As our experimental results show, our main idea of using a compact feature representation to avoid very high dimensional representations and thus an increased processing time seems feasible. The novel frequency-based features we introduced also contributed to a good overall classification performance.

2.5 Summary of Thesis results

The main results of this chapter can be summarised as follows. For NER in Hungarian the author participated in the creation of the first Hungarian NER reference corpus which allowed researchers to investigate statistical approaches to Entity Recognition in Hungarian texts. This is a joint, inseparable contribution between the authors of [18] and the linguist colleagues who carried out the annotation work of the corpus. Together with his colleagues, the author designed a suitable feature representation for training machine learning models and set up an efficient learning model on the corpus that achieved a phrase level F measure performance of 94.76%. In the construction of the Named Entity Recognition system, the author made major contributions in designing the feature representation for learning algorithms. The most important useful

[5] Consider, for example, the token *Ford*. This can be either a location (an airport), a person's name (Henry Ford), an organization (the Ford company) or a miscellaneous entity (the automobile brand Ford)

characteristics of the feature representation are the following:

- diversity: The author incorporated an extensive feature representation that benefits from all the most successful features arising from the surface, orthographical and morpho-sytactic levels described in the literature.
- compactness: Using the above-mentioned compact representation the dimensionality of the feature space remained moderate and permitted the fast training and testing (processing) of boosted C4.5 decision tree learning algorithms.
- novelty: The author introduced some novel frequency-based features that contributed to the performance of the NER system.

Chapter 3

English Named Entity Recognition

In this section we describe the experiments we carried out in English newswire NER. Using the same model as that outlined for the Hungarian problem, we will show that our representation is suitable for NER in various languages, i.e. language independent NER models can be built on the basis of our representation. Different domains of application suggest some extensions to the feature set described for Hungarian, and these extensions can definitely improve the overall recognition accuracy, as we will show.

3.1 The corpus

For the experiments in English NER we used a publicly available dataset, which is a segment of the Reuters corpus [49] that contains news stories from the Reuters News Agency from 1996-1997. A small part of this corpus was annotated with NE labels for the NER shared task of the Computational Natural Laguage Learning (CoNLL) conference in 2003. Since we used the same design as this corpus when building a reference corpus for Hungarian, it was quite rapid and easy to adapt our system to the English NER task.

The dataset consists of a training part, a development part to tune system parameters on, and a test part for evaluations. Here the test dataset has several different characteristics from the training and development sets, which makes the dataset more challenging.

The format of the corpus is similar to that outlined for the Hungarian dataset. It is:

U.N. NNP I-NP I-ORG
official NN I-NP O
Ekeus NNP I-NP I-PER
heads VBZ I-VP O
for IN I-PP O
Baghdad NNP I-NP I-LOC
. . O O

The main characteristics of the dataset are summarised in the following two tables[1]:

	Sentences	Tokens
Training set	14987	203621
Development set	3466	51362
Test set	3684	46435

Table 3.1: Number of articles, sentences and tokens in the corpus.

	LOC	MISC	ORG	PER
Training set	7140	3438	6321	6600
Development set	1837	922	1341	1842
Test set	1668	702	1661	1617

Table 3.2: Number of named entities per data file.

3.2 Description of our model extensions and results

The slight differences between the Hungarian and the English texts we used motivated us to develop some additional features that allowed classifiers to overcome such challenges that were specific to the English problem, and were impossible to capture through the original feature representation. Here we will present the additional features one by one along with the motivation for adding them to the model.

- **Topic code:** The English dataset consists of stories with various topic/domain, e.g. sports, economics, political news, while the Hungarian corpus was homogenous in this sense, that it contained just short business news texts. The domain of a text has a major impact on the distribution of Named Entities in the text. Different NE classes appear more frequently in one domain than another (e.g. location names appear frequently in tour-guide texts, while being rare in sports news). The domain also has an impact on the behaviour of certain NE phrases. For example, city names are typically tagged as locations, but they also appear as organisation names in sports news, where city names are frequently used to refer to the city's sport clubs. These facts mean that it makes sense to add a feature that permits the topic differentiation of articles. Here we classed articles as political, financial or sports news texts.

[1] This example and the tables both come from Tjong Kim Sang and De Muelder's shared task paper [7].

- **Document zone:** Certain document zones can easily be identified in Reuters articles, and the surface form of texts (and thus NEs) in different zones can vary. For example, headlines are written in capitalised form, which causes most of the surface features to behave in a different way. This can result in erroneous classifications in headlines and also in lower performance in flaw text, as some important features behaved unreliably throughout the whole corpus. Adding further attributes that distinguish zones by taking different values in different parts of a typical article offers the chance of learning complex decision rules (e.g. that it worths using capitalisation info in flaw text, while it is better to rely on other feature types like sentence position info instead of capitalisation in headlines). An alternative solution is to try restoring the original capitalisation of the text [50]. We decided to distinguish zones with additional features (we differentiated between headline, dateline and article body zones) and left the problem of inducing patterns for recognition in headlines to the learning models.

- **syntactic / morpho-syntactic description:** The organisers of the CoNLL-2003 NER challenge provided automatically generated part-of-speech and syntax chunk codes with the texts. POS codes can help in identifying NEs, while chunk codes might be useful in determining boundaries for longer name phrases. Even though these features would clearly benefit NER, we should note here that these have to be generated automatically by POS taggers and syntax chunkers. This also means that these attributes are 'burdened' with a certain amount of noise (i.e. the error rate of the system that generates feature values). Since NER-relevant information in POS codes (or similar info) can also be deduced from other simpler surface features, while the current state-of-the-art accuracy of chunking is hardly better than a NER tagger's performance hence the use of generated tags is questionable in this case. Taking into account the fact that the use of such deep knowledge features would detriment the adaptability and flexibility of NE recognisers, as the development of POS taggers or chunkers is resource-intensive and costy, it really makes sense to exclude such attributes from NER systems [2]. Using syntactic and grammatic roles as features would be particularly useful when the classification scheme follows the semantics of NEs, i.e. when metonimies are to be resolved [35] [38].

- **corpus/web statistics for lemmatisation:** Sometimes Named Entities appear with affixes in the text and such examples are harder to classify properly than the official forms (lemmas) of proper names. Affixes can also help disambiguation if we seek to distinguish metonymic and literal uses, as for example, company names appearing in plural usually refer to a brand or product of the company in question. Take, for example, *Little old ladies in small Renaults*$_{ORG-FOR-PRODUCT}$. Frequency statistics gathered from large corpora or the Internet can be utilised

[2] Only features that were found useful in one task should be carried over to later tasks/experiments. This conclusion led us to discard deep knowledge features from our models and we did not try using such features in later tasks in different scenarios.

to lemmatise proper names, to extract very limited morpho-syntactic information (like plural form) and to separate individual NEs that follow each other in the text without any separating punctuation. Heuristics based on these statistics exploit the hypothesis that any Named Entity appears in a corpus of proper size (here the WWW can be regarded as a corpus that is practically infinite in size) and all entity names appear in normalized form a magnitude more frequently than in affixed form. These heuristics will be elaborated in the next section.

3.3 Frequency heuristics for NE-normalisation

3.3.1 Separating consecutive NEs

In the majority of cases, consecutive Named Entities either follow each other with a separating punctuation mark (enumerations), or belong to different classes. In the first case, a non-labeled token separates the two phrases, while in the second case the different class labels identify the boundaries. Rarely do two or more NEs of the same type appear consecutively in a sentence. In such cases the phrasal boundaries must be marked (we used a "B-" prefix for phrase starting tokens).
An example of consecutive NEs of the same type is:

- *The Russians, working for the Aerostan firm in the Russian republic of Tatarstan, were taken hostage after a* ***Taleban***$_{MISC}$ | ***MiG-19***$_{MISC}$ *fighter forced their cargo plane to land in August 1995.*

Sometimes separating punctuation is absent from the text and this results in consecutive NEs, as in the following example:

- ***Buenos***$_{LOC}$ ***Aires***$_{LOC}$ | ***Quequen***$_{LOC}$ | ***Rosario***$_{LOC}$ | ***Bahia***$_{LOC}$ ***Blanca***$_{LOC}$

We used Wikipedia to identify phrase boundaries. As we had too few consecutive NEs in the CoNLL corpus to perform a reliable evaluation of a corpus-based splitting heuristic, we removed all punctuation marks from the corpus. This made consecutive NEs much more frequent in the text. We then queried Wikipedia for all entities that had two or more tokens. If we found an article sharing the same title as the whole query, or the majority of the occurrences of the phrase in the Google snippets occurred without punctuation marks inside, we treated the query phrase as a single entity. If a punctuation mark was inside the phrase in the majority of the cases, we separated the phrase at the position of the punctuation mark. This method allowed us to separate phrases like *Golan Heights* | *Israel*. If there was no hit for the query in Wikipedia, but we were able to find a wiki entry for two or more parts of the query, we put phrase boundaries following the Wiki entries. This way we successfully identified phrases like *Taleban* | *MiG-19* and many enumerations that lacked the separating commas due to the removal of punctuation marks from the data. We also made use of a first names list here containing 3217 first names, which allowed us to avoid the erroneous separation of full names (first name, last name pairs). Of course, a more comprehensive first names list would have been useful to us. Our system suffered from the absence of Romanian and Arabic first names here. This heuristic improved the overall performance of the NER tagger on data that lacked punctuations by a significant 1.42% (or 8.1% error reduction). The heuristic itself managed to recognize the 'B-' (NE-starting token that is preceded by another NE of the same type) tags with an $F_{\beta=1}$-measure of 75.19% (precision 71.7%; recall 79.03%).

We should also mention here that some of the 'B-'-tagged phrases in the CoNLL database are arguably consecutive NEs, but are actually single entities (e.g. phrases like *English Moslems* or 'City State' phrases like *Rochester NY*). Our heuristic does not separate such cases as they usually seem to be single NEs for the online encyclopedia - and they can be treated as single entities as well in an Information Extraction system. Without such cases the recall score of our system would have been even higher.

3.3.2 NE lemmatisation and morpho-syntactic analysis with web-search based heuristics

Sometimes Entity names appear with affixes in plain text, like *s* for plural form or *'s* for possession. Splitting off these affixes can be useful for many reasons. E.g.

- when acquiring the lemmatised form of the NE (morpho-syntactic analysers are not well-suited for proper names and often fail to provide lemmas of NEs)
- when analysing affixes for use as features for particular applications.

The lemmatisation of named entities can be really helpful for dictionary lookup, while affixes can be beneficial to classification performance as features when semantic categorisation is required (e.g. the plural form of organisation names strongly indicate org-for-product type metonimies). However, it is often not straightforward to decide whether an NE is in a lemmatised or affixed form. Take, for instance, the following samples:

- *Epson's /affixed/, Ford's /ambiguous/, McDonald's /lemma/,*
  Sotheby's /lemma/
- *Renaults /affixed/, Philips /ambiguous/, Advanced Micro Devices /lemma/*

Above we have outlined a general issue of removing affixes from Named Entities, a problem that appears in English, but is far more important in agglutinative languages like Hungarian. We proposed a solution identifying the pluralty/singularity of organization names in [38], where the task was the classification of organisation names to literal and different types of metonimic usages. Using the name of an organisation to denote its product is a common phenomenon in language and in such uses organisation names can appear in plural form as well. Hence identifying the pluralty of an ORG name is a helpful feature for identifying org-for-product metonimies. In [38] we made the assumption that only ORG names ending with the letter *s* are potential plural forms and used such names in the training data to determine a threshold of the frequency ratio of the word form and the possible lematised form which could separate plural proper names from lemmas ending with letter *s*. We applied the calculated threshold to determine the singular/plural status of NEs in the test corpus in order to assign values to our plural feature. This heuristic performed with 100% accuracy on the Semeval-2007 metonymy resolution shared task corpus and benefited the recognition of org-for-product metonymies.

Assigning values to our plural feature motivated our idea to use web frequency counts for NE lemmatisation. We judged the accuracy we obtained for the values of the plural feature to be quite promising and later on we addressed NE lemmatisation as a standalone task. The evaluation we carried out for both English and Hungarian (again, to demonstrate the language independent nature of the approach) demonstrate that accurate heuristics can be developed for removing inflectional suffixes using corpus or web frquency information. We consider this finding a valuable result since morphological analysers (that are the general tools used for obtaining the lemmas of common words) are not well suited for lemmatising Named Entities. Morphological Analysers usually rely on an exhaustive list of possible lemmas – a resource that is impossible to gather in the case of Named Entities – and thus they do not perform well when used for providing lemmas and inflectional affixes of names. A thorough analysis of these issues is discussed in detail in [37].

The corpora

The lists of negative and positive examples for lemmatisation were collected manually. We adopted the principal rule that we had to work on real-world examples (we did not generate fictitious examples), so the annotator team was asked to browse the Internet and collect "interesting" cases. These corpora are the unions of the lists collected by 3 linguists and were checked by the chief annotator. The samples mainly consist of person names, company names and geographical locations occurrences on web pages. Table 3.3 lists the size of each corpora (constructed for the three problems). The corpora are accessible and are free of charge.

A problematic case of finding the lemma of a NE is when the NE ends in an apparent suffix (which we will call the *suffix* problem in the following). In agglutinative languages such as Hungarian, NEs can have hundreds of different inflections. In English, nouns can only bear the plural or the possessive marker *-s* or *'s*. There are NEs that end in an apparent suffix (such as *Adidas* in English), but this pseudo-suffix belongs to the lemma of the NE and should not to be cut off.

We decided to build two corpora for the suffix problem; one for Hungarian and one for English and we devised the possible suffix lists for the two languages. In Hungarian more than one suffix can be matched to several phrases. In these cases we examined every possible cut and the correct lemma (chosen by a linguist expert) became a positive example, while every other cut was treated as a negative one.

	Eng suffix	Hun suffix
positive examples	74	207
negative examples	84	543

Table 3.3: The sizes of the corpora.

The feature set

To create training datasets for machine learning methods - which try to learn how to separate correct and incorrect cuts based on labeled examples - we sent queries to the Google and Yahoo search engines using their APIs . The queries started and finished in quotation marks and the site:.hu constraint was used in the Hungarian experiments. In the suffix tasks, we sent queries with and without suffixes to both engines and collected the number of hits. The original database contained four dimensional feature vectors. Two dimensions list the number of Google hits and two components list similar values from the Yahoo search engine.

Our preliminary experiments showed (see Table 3.4 for English and Table 3.5 for Hungarian below) that using just the original form of the datasets for the suffix tasks is not optimal in terms of classification accuracy. Hence we performed some basic transformations on the original data. First we experimented with feature sets where only one of the two search engines' hits was present (first column). Second, the first component of the feature vector was divided by the second component. If the given second component was zero, then the new feature value was also zero (second column). This yielded a one dimensional dataset for the two individual search engines and a two dimensional one when we utilized both Yahoo and Google hits for the classification task.

NE lemmatisation results

In this task the training algorithms achieve their best performance on the transformed datasets using rates of query hits (this holds when Yahoo or Google searches were performed). One could say that the rate of the hits (one feature) is the best characterisation in this task. However, we can see that with the Hungarian suffix problem the original dataset characterises the problem better, and thus the transformation is really unnecessary. The best results for the Hungarian suffix problem are achieved on the full dataset, but they are almost the same as those for untransformed Yahoo dataset. Without doubt, this is due to the special property of the Yahoo search engine which searches accent sensitively, in contrast to Google. For example, for the query "Ottó" Google finds every webpage which contains Ottó and Otto as well, while Yahoo just returns the Ottó-s.

Search Engine	Feature set	kNN ($k = 3$)	C4.5	MaxEnt	Baseline
Both	freq counts	89.24	86.71	84.81	53.16
	freq rate	93.04	91.77	73.42	53.16
Google	freq counts	87.34	87.34	82.28	53.16
	freq rate	93.67	87.97	90.51	53.16
Yahoo	freq counts	89.87	86.08	84.18	53.16
	freq rate	91.77	87.34	88.61	53.16

Table 3.4: Suffix task results for English obtained by applying different learning methods.

Search Engine	Feature set	kNN ($k = 3$)	C4.5	MaxEnt	Baseline
Both	freq counts	94.27	82.67	88.27	72.40
	freq rate	84.67	81.73	72.40	72.40
Google	freq counts	85.33	82.40	83.33	72.40
	freq rate	83.60	83.60	77.60	72.40
Yahoo	freq counts	93.73	83.87	86.13	72.40
	freq rate	87.20	87.20	74.00	72.40

Table 3.5: Suffix task results for English obtained by applying different learning methods.

3.4 Results

The results that are counted as the author's own contribution are presented above. Here we summarise the overall results of our systems as a whole, which is a joint contribution of the co-authors of [33] [34]. Using the exact-match evaluation criterion of the CoNLL conference NER shared tasks, our model achieved an $F_{\beta=1} = 89.02\%$ and $F_{\beta=1} = 91.41\%$ in combination with other top performing systems of the CoNLL-2003 conference. These results are competitive with those of the current state-of-the-art systems.

In metonymy resolution, a system developed in collaboration with researchers of the Technical University of Budapest [38] achieved an overall accuracy of 72.80% for organisation name metonymies and 84.36% for location names. These results are also competitive those of other current systems. The fact that these performance scores are only marginally better than simple baseline performances shows that the task of resolving figurative language (metonymies) is rather complex.

3.5 Comparison and conclusions

3.5.1 Named Entity Recognition

Previous research on English NER dates back to the early 90s. The article by Lisa F. Rau [51] from 1991 is often cited as the first paper on this topic. The intensity of research on NER was boosted by the Message Understanding Conferences, the first major series of events in part dedicated to the task in the 90s [14],[6]. In the past two decades, machine learning solutions to the NER problem have become dominant, at least when the number of category types to recognise are restricted. Open domain NER systems [52], [53], [54] and NER systems for much more but restricted number of types [55], [56] are not directly comparable to our results. A shared task challenge of the CoNLL conference in 2003 was dedicated to multilingual named entity recognition, with English being one of the evaluation languages. Since our results follow the CoNLL conference standards, systems of the 2003 shared task are the best candidates for comparison[3].

System	phrase-level $F_{\beta=1}$
Florian et al. [57]	88.76%
Chieu and Ng [58]	88.31%
Klein et al. [59]	86.07%
Zhang and Johnson [60]	85.50%
Carreras et al. [61]	85.00%

Table 3.6: Performance of the five best systems at CoNLL-2003.

The authors of the shared task description paper report an $F_{\beta=1} = 90.30\%$ for the majority-voting based combination of the 5 best performing systems. Using our model as one in the voting scheme yields an $F_{\beta=1} = 91.41\%$, which shows that our model exhibits some different characteristics from the participating ones that had a positive impact on the overall performance of a voting model.

Summarising our findings, we think that the feature representation presented in the previous chapter that was designed for Hungarian is capable of solving NER language independently as long as the resources (dictionaries, lists, frquency data, etc.) for a new languages are on hand. With some extensions that were discussed in this chapter and addressed the most obvious differences between the English and the Hungarian dataset (like multiple domains, document zones, etc.) the performance of models trained on the feature representation introduced here are comparable with those of the best current systems. They therefore hold great promise for the future in tackling various NER tasks.

3.5.2 NE lemmatisation

NE lemmatisation has not attracted much attention so far because it is not such a serious problem in major languages like English and Spanish as it is in agglutinative languages. An expert rule-based and several string distance-based methods for Polish person name inflection removal were introduced in [62]. A corpus-based rule induction

[3] To the best of the author's knowledge, no significantly better results using the same standards have been published since, so we restrict ourselves to a comparison with shared task systems.

method was studied for every kind of unknown words in Slovene in [63]. The scope of our study lies between these two as we deal with different kinds of NEs.

Due to the above reasons it is not straightforward to compare our results for NE lemmatisation with accuracy scores reported for different approaches. On the other hand, regarding the high performance scores we obtained and the fact that our method uses no specific resources (like lists of lemmas) we consider our results very promising.

3.6 Summary of Thesis results

The main results of this chapter can be summarised as follows. The author participated in adapting a NER system designed for the Hungarian language to a similar task in English. Together with his colleagues, the author extended the feature representation for training machine learning models and used the same, efficient learning model that was introduced for Hungarian NER. This system attained a phrase level F measure score of 89.02%.

The author also participated in the development of a MaxEnt-based system for the metonymy resolution shared task of SemEval-2007 [35]. In the NE-metonymy classifier that was entered the challenge by the author and his colleagues, the web-based approach described in this chapter designed to remove inflectional affixes from Named Entities was used successfully as a feature to classify org-for-product metonymies.

When constructing the English Named Entity Recognition system, the author made major contributions in designing the feature extensions for learning algorithms. When developing the metonymy resolution system, the author made major contributions in designing the frequency-based heuristics for NE lemmatisation and feature generation from NE affixes (plural feature in [38]). In [37] the author's contribution is the idea and general concept of using web frequency counts for Named Entity lemmatisation (NE normalisation or affixes as features for other tasks), while the experiments were actually carried out by the co-authors.

Chapter 4

The anonymisation of Medical Records

In the human life sciences, the NER task is crucial because a de-identified text can be made publicly available for non-hospital researchers to facilitate research on human diseases. However, the records of patients include explicit personal health information (PHI), and this fact hinders the release of many useful data sets because laws relating to data protection and personal patient rights forbid this. According to the guidelines of Health Information Portability and Accountability Act (HIPAA) of the US, the medical discharge summaries released must be free of seventeen categories of textual PHI, among which the following actually appear in discharge summaries:

- first and last names of patients, their health proxies and family members
  e.g.: *Mrs.* ***[Mary Joe]*** *was admitted...*;
- doctors' first and last names
  e.g.: *He met with Dr.* ***[John Bland]****, MD.*;
- identification numbers
  e.g.: *Provider Number:* ***[12344]****.*;
- age (above 89 years)
  e.g.: *She is a* ***[91yo]*** *lady with congestive heart failure.*;
- telephone, fax, and pager numbers
  e.g.: *Please call Dr. Cornea at* ***[555-23456]***;
- hospital names
  e.g.: *The patient was transferred to* ***[Gates 4]****.*;
- geographic locations
  e.g.: *He lives in* ***[Newton]****.*;
- dates (excluding years)
  e.g.: *ADMISSION DATE:* ***[10/29]****/1997.*.

Removing such items of PHI is the main goal of the de-identification process. Anonymization goes one step beyond the removal of personal information and attempts to identify and classify personal information in the text to one of the HIPAA-defined categories. This categorisation permits the replacement of personal data instead of simple deletion, and it has one major advantage: the replacement of PHIs with artificially generated realistic substitutes not only preserves the readability of text, but also the artificial substitutes actually disguise those very few items of personal information that remain in the document (the reader will never know whether a single label was the original or a substitute).

In this section we will describe our experiments on the adaptation of our NER model to a new domain, that of medical discharge summaries. The anonymisation of medical texts was the focus of a shared task challenge organised by I2B2 (Informatics for Integrating Biology and the Bedside) in 2006. The anonymisation task is very similar in nature to the NER task for newspaper texts. Here the aim is to find any items of information that could enable one to identify a patient – contrary to data protection laws and personal rights – and thus hinder sharing medical documents for research purposes. Ignoring such complex cases when the meaning of the text itself can identify the patient (e.g. *His brother is a quarterback of the Saints*), personal information is mainly expressed by named entities. Making this straightforward simplification (assuming that medical documentation is free of literal texts like the example above), the de-identification shared task challenge focused on the detection of explicit personal health information items (PHIs) in discharge summaries.

Since explicit PHI is mostly expressed in terms of NEs, the adaptation of existing NER systems to the medical domain seemed a promising way to achieve good performance scores with a moderate development cost.

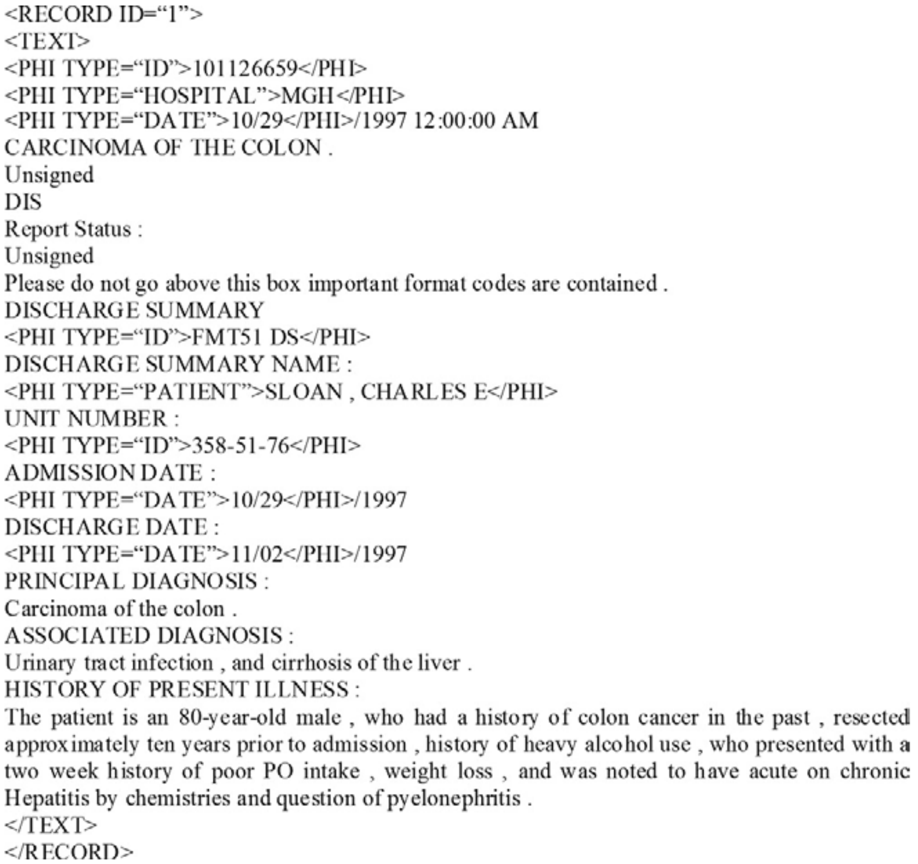
The rest of this section is organised as follows. We briefly describe the datasets we used, explain the adaptation process of our system and summarise the main findings of the shared task challenge. The author designed and implemented several extensions of the previously described NER models, relating to the feature space representation used. These extensions exploit the specific characteristics of medical texts and have a beneficial impact on the accuracy of de-identification. All modifications relating to the feature space representation are counted as the author's own contributions to this project.

4.1 The corpus

In the experiments described here we used the de-identification corpus prepared by researchers of the I2B2 consortium www.i2b2.org for the de-identification challenge of the 1st I2B2 Workshop on Natural Language Processing Challenges for Clinical Records. The dataset consisted of 889 annotated discharge summaries, out of which 200 randomly selected documents were chosen for the official system evaluation. An important characteristic of the data was that it contained re-identified PHIs. Since the real personal information had to be concealed from the challenge participants as well, the organisers replaced all tagged PHI in the corpus with artificially generated realistic surrogates. Here realistic means that they sought to preserve the surface characteristics of the PHI tokens, and also to maintain coreference relations between phrases. This means that if a patient name (*John F. Smith*) was found in the text, it was replaced by a generated name with similar orthography (e.g. *Samuel L. Taylor*). Then subsequent references of the same entity (e.g. *J. F. Smith*) were replaced by the same surrogate (in this case, *S. L. Taylor*). Since the challenge organisers wanted to concentrate on the separation of PHI and non-PHI tokens, they made the dataset more challenging with 2 modifications during the re-identification process:

- They added out-of-vocabulary surrogates to force systems to use contextual patterns, rather than dictionaries.
- They replaced some of the randomly generated PHI surrogates (patient and doctor names) with medical terminology like disease, treatment, drug names and so on. This way systems were forced to work reliably on challenging ambiguous PHIs.

A small sample of the corpus and a table with its main characteristics can be seen below. For more details about the corpus, the annotation process, levels of ambiguity, etc., see [11].



```
<RECORD ID="1">
<TEXT>
<PHI TYPE="ID">101126659</PHI>
<PHI TYPE="HOSPITAL">MGH</PHI>
<PHI TYPE="DATE">10/29</PHI>/1997 12:00:00 AM
CARCINOMA OF THE COLON .
Unsigned
DIS
Report Status :
Unsigned
Please do not go above this box important format codes are contained .
DISCHARGE SUMMARY
<PHI TYPE="ID">FMT51 DS</PHI>
DISCHARGE SUMMARY NAME :
<PHI TYPE="PATIENT">SLOAN , CHARLES E</PHI>
UNIT NUMBER :
<PHI TYPE="ID">358-51-76</PHI>
ADMISSION DATE :
<PHI TYPE="DATE">10/29</PHI>/1997
DISCHARGE DATE :
<PHI TYPE="DATE">11/02</PHI>/1997
PRINCIPAL DIAGNOSIS :
Carcinoma of the colon .
ASSOCIATED DIAGNOSIS :
Urinary tract infection , and cirrhosis of the liver .
HISTORY OF PRESENT ILLNESS :
The patient is an 80-year-old male , who had a history of colon cancer in the past , resected
approximately ten years prior to admission , history of heavy alcohol use , who presented with a
two week history of poor PO intake , weight loss , and was noted to have acute on chronic
Hepatitis by chemistries and question of pyelonephritis .
</TEXT>
</RECORD>
```

Figure 4.1: Sample discharge summary.

4.2 Evaluation Methods and Preliminary Experiments

In our experiments we performed two different evaluation methods, namely a token level 8-way and a 9-way F measure. The 8-way evaluation excludes non-PHI true positives and thus measures the performance of identifying the 8 PHI classes, while the 9-way evaluation takes into account non-PHI class as well. The latter metric examines the correct recognition of non-PHI, because this class is important for preserving the document's information content. The 9-way F measure was the official evaluation metric used for the I2B2 challenge. In the Results section we apply an 8-way evaluation to see how well different models recognise PHI tokens, while the 9-way F measure is more suitable and used for a general comparison of system performance. Other shared tasks on NER-like problems used phrase-level evaluation metrics that are better suited for other Information Extraction tasks. For de-identification token-level evaluation is more appropriate, as the partial removal of a PHI should receive a partial credit, instead of a full penalty.

We should also mention here that the evaluation script we used implemented an equal-weighted F measure ($F_{\beta=1}$). This is probably not the most suitable evaluation method for the de-identification of medical records, as the removal of all PHIs is extremely important, so perhaps recall should be given a higher priority. Moreover, the failure of the removal of one PHI or another PHI is often not of the same degree of seriousness (consider the failure of the removal of a patient's family name or a small part of a hospital name like "of" in the document-the former seriously conflicts with the HIPAA guidelines, while the latter does not). Thus while it is not straightforward to give an ideal evaluation metric for the de-identification task, we think the evaluations used here are still good indicators of the quality of our results.

We evaluated two baseline methods for comparison, in order to get a better insight to the value of our results. These baseline systems are a majority baseline and a simple decision tree classifier.

Majority class: This simple baseline predicts non-PHI for each token (most frequent class).

C4.5: We used a single C4.5 learner instead of AdaBoostM1 and C4.5, with token triggers.

Excluding all domain specific extensions that we implemented, our model yields an F measure score of 99.48% in 9-way evaluation, and thus outperforms the mean F measure of the systems (99.19%) submitted to the competition. We consider this a valuable result as this system exploited none of the special characteristics of medical texts described earlier. In 8-way evaluation this system got a 94.34% F score, while our second baseline method (a C4.5 with the domain extensions but without boosting) achieved a 94.93% F measure score. This shows how important it is to exploit the special characteristics of the medical domain texts.

4.3 Novel features for the de-id task

We extended the representation designed for newswire NER systems with two additional features that utilised the special characteristics of medical discharge summaries and the de-identification task. These additional features enabled us to fine-tune our model for the new application domain and achieve peak performance.

4.3.1 Different use of trigger words to describe local context

The use of trigger words is not straightforward, so we used them in three different ways in our experiments: we collected the three preceding and three subsequent tokens of all tagged tokens in the train set (we refer to this feature set as the token trigger later on); similarly, we collected subsequent tokens of tagged phrases and used a wider window for this feature (phrase trigger); and then we collected the bi- and trigrams around the phrases of the train texts (trigram trigger).

In the case of **token trigger**ing, we collect this kind of information for all tagged tokens, not phrases. This way "M.D." should be a similarly strong trigger for the DOCTOR class with offset 2. Furthermore, it becomes a somewhat weaker token-trigger for the DOCTOR class with an offset of 3 /it typically appears with 3 offset to DOCTOR tokens (like "John Smith, M.D.") and to non-PHI tokens (as in "visited Dr. Smith, M.D.")/.

Phrase trigger means the kind of tokens that appear before or after a complete PHI phrase (perhaps several tokens long). For example, "M.D." is a trigger for DOCTOR class phrases with an offset of 2, as the usual pattern in text is "DOCTOR_NAME, M.D.". Of course, as the classification itself is performed by a token-level model, this feature helps us to identify just the first or last token of a doctor name (depending on the sign of the offset). In "John Smith, M.D." only the instance for token "Smith" has this feature set to true.

Bigram/trigram triggers do not collect single trigger tokens (which are good predictors of a certain class label), but 2 or 3 token-long sequences. In this model, ", M.D." should be a strong indicator of the DOCTOR class, not "," with offset 1 or "M.D." with an offset of 2 on their own.

The collected trigger lists for each of the three cases were filtered according to their frequency and information gain on the class labels (that is, according to their predictive power). A significant difference in the predictions was noticed in the experiments where only the use of triggers was altered; hence we decided to combine their forecasts to exploit their advantages better. Even if only marginal improvements could been achieved this way, this kind of voting strategy improved overall performance compared to that when using the best one of the three different trigger features alone.

4.3.2 Regular expressions for well formed classes

Among the eight classes of entity types present in the de-identification challenge corpus there were several classes with instances that followed very strict, steady surface patterns. These classes included

PHONE numbers We added simple regular expressions that identify phone number with/without area codes and extensions, pager numbers, etc. ide jönnek pédák

AGE phrases We added regular patterns of how patient age might be expressed in natural language. ide jönnek pédák

DATES We added patterns that trigger the most common date and time formats found in the text. ide jönnek pédák

LOCATIONS (full addresses) We added patterns that matched common longer geographical place names. The most typical examples here were "city, state" phrases and "street, city, state, area code" phrases. These patterns used dictionaries to match only valid state names as real US state names were found in the re-identified text in location phrases. ide jönnek pédák

IDs The IDs in the text followed very strict positional and surface rules that have been captured by other features developed earlier for newswire NER (sentence position, number, contains digit, etc). Because of this, we added no further regular expression patterns to cover ID instances as the out-domain model also learned the surface patterns perfectly.

Selection of regexp features

The above-mentioned regular expression patterns were constructed manually, using the lists of entites found in the training dataset for each class. We defined patterns that matched a reasonable amount of target phrases. As we ignored the manual validation of the patterns defined, this procedure took just 2 hours. Later on all the proposed regexp/class pairs were automatically validated against the whole training corpus and we calculated the precision and coverage for each pattern. Only the most reliable (very high precision) patterns were added to the learning system as regexp features. This way we managed to add very strong features to the learning model which obviously learned to recognise these well-formed entity phrases based on the binary features corresponding to one of the matching regular expressions. On the other hand, adding these expressions to the learning model as features (instead of applying the most reliable ones as post processing rules) made it possible for the machine learning model to induce contextual rules where the regexp features fail. As the regexp features were developed using just the target phrases, sometimes these regular patterns overfitted in some typical situations. This way we managed to reduce the risks of overfitting. Trying to cover entity classes using the regularities of their surface characteristics still holds the risk of ovefitting the model on the training dataset. If a pattern worked with 100% precision and a relatively high coverage (like the regexp *XXX-XXX-XXXX* for phone numbers, where Xs stands for numeric characters), the system would obviously learn to rely entirely on the single feature and discard all contextual information. If entities in previously unseen data shared different surface characteristics, then they would not be recognized this way. This happened with phone numbers in the challenge test dataset (some phone number contained spaces, while in the training dataset only dashed phone numbers were present).

4.3.3 Document section headline information

Medical discharge summaries are semi-structured documents, consisting of many typical sections, identified by titles. However, these headings probably vary from hospital to hospital and - as the challenge data suggested - none of them are required to be present in a discharge summary. On the other hand, these headings are very useful sources of information for entity tagging as the relative entropy of tokens decreases significantly in free-text parts belonging to certain headers.

We identified those lines as common headings of the challenge dataset which appeared in at least 15 records (reasonable frequency) and ended either by ":" or "*****".

This regularity of line ending allowed us to avoid the identification of the most frequent common lines as headings. This way, heading information was collected in a fully automatic way (no manual evaluation/supervision). Hence we think similar headings could be captured in texts originating from different health institutes with similar simple methods.

Header features

We added the actual text heading to the learning model as a feature for each instance. This way the model could differentiate between similar instances depending on the section in which they appeared. The novel characteristic that differentiated this feature from other similar ones (like document/sentence position, preceding words) was that the same code was assigned to a varying number of tokens (up to the appearance of the next common header).

Header-based post processing

We added post processing rules to overwrite PHI tokens in sections where the entropy of PHI tokens was 0 (only PHIs of one class appeared in that header). This post-processing allowed us to avoid obvious mistakes when the classification failed due to the incidental effect of some other feature. Some of the section headings were so reliable that they led us to discover annotation errors:

The sections below the ******* discharge orders ******* heading contained 102 patient names, 8 doctor names and 110 IDs. The eight doctor names here were probably due to annotation errors.

4.3.4 Dynamic feature generation using headings (iterative learning)

We exploited the trusted information on PHI categories in certain document sections (following reliable headings) in a third way. The motivation here was that - apart from rare cases - a single token usually belongs to the same class if it appears several times in a document. This phenomenon is referred to as label consistency, which means that at the document level, the labeling of terms is usually consistent. To a limited extent, label consistency might hold at the corpus level as well. Hospital names for example should be tagged consistently throughout the whole corpus, if the text originates from a single source. Our method exploited document level label consistency in the following way:

1. First, we trained a model and tagged all documents for PHIs
2. Second, we performed the following steps iteratively:
   - Based on observations of the training corpus we collected each tagged PHI item from sections following reliable headings
   - We formed lists of the tagged trusted PHIs and added them as dictionaries to the model and retrained the system
   - We tagged all documents for PHI items using the retrained model. If this new model tagged sufficiently more trusted PHIs than the previous one, we repeated the procedure of step 2.

This iterative approach took advantage of possible knowledge about the token's tagging in simple contexts in order to tag the token in more ambiguous text parts. This dynamic feature generation approach gave an overall improvement of 0.33% (from 96.38% to 96.71% in an 8-way F measure). The post-processing methods we applied slightly reduced this gain, but we think this might be a helpful approach where the label consistency hypothesis can significantly contribute to the performance of statistical models.

4.4 Analysis of the Feature Set

The general purpose feature set we used is briefly described in Section 2.2.2. Here our 138 attributes had different benefit on the overall performance. The lists collected (from the Internet, the training set and from the CoNLL-2003 dataset), for example, had no positive impact on the model as later experiments showed. In particular, the two lists containing typical non-entity elements (one containing non-PHIs and one containing non-NEs from the out domain NER corpus) only confused the model and lowered the classification accuracy a little bit. It is also a somewhat surprising result that a list of first names brought no benefit to the model, although this gazetteer proved extremely helpful in our previous studies. Of course, the re-identifiedd characteristic of the I2B2 dataset captures this fact: name phrases in the I2B2 dataset were often replaced by out-of-vocabulary words or typical non-name words (diseases, for example).

For an analysis of the effectiveness of our features, we divided them into ten subsets, grouping those of a similar type. These subsets of features were added to the feature pool in a greedy way (most useful first, i.e. the one that gave the highest improvement in terms of classification performance) in order to evaluate their contribution to the system's performance.

The groups of features included in order of significance were the following (see Figure 2):

1. Basic features: initial letter type, trigger, predictions for previous tokens
2. Orthographical features
3. Frequency information
4. Document heading information
5. Regular expressions for well-formed classes
6. Location dictionaries (countries, cities)
7. Sentence position information
8. The word is inside quotation marks/brackets
9. First names list
10. Gazetteers of non-PHIs

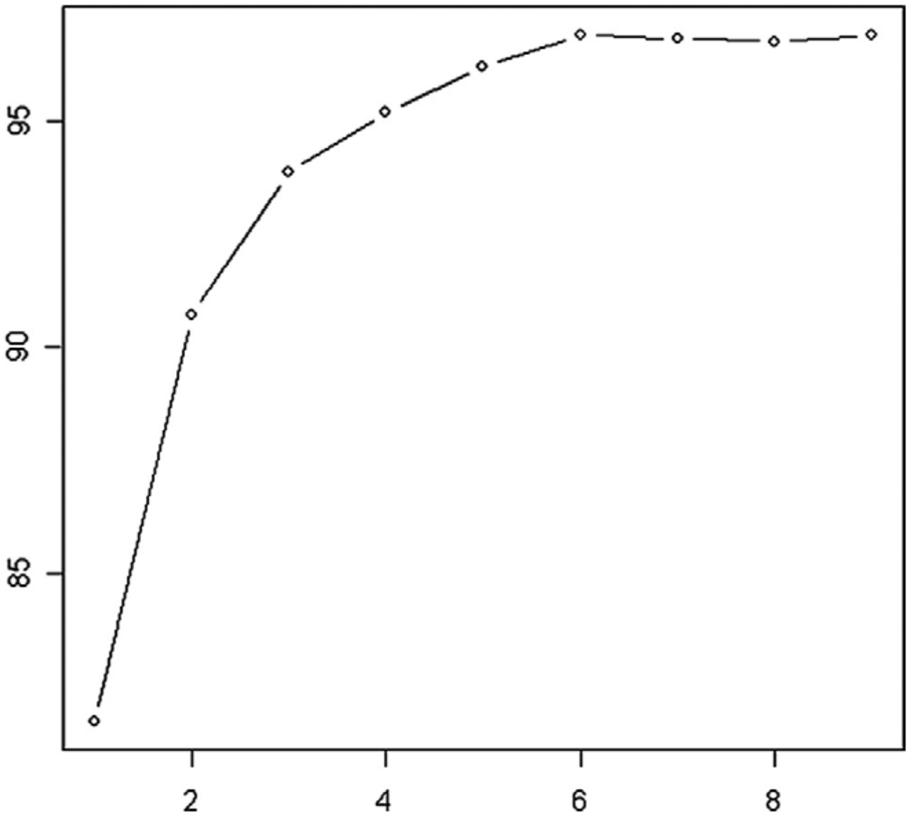


Figure 4.2: System performance with features of different types added to the system. The evaluation is an instance-level $F_{\beta=1}$ (that is, CoNLL-style evaluation).

4.5 Results

The results that are counted as the author's own contribution are given above. Here we summarise the results of our system as a whole, which is a joint contribution of the developers of the system (the authors of of [39]). Using the standard evaluation criteria of the I2B2 workshop, our model achieved a 99.7534% token level accuracy (9-way) and 98.0% token level $F_{\beta=1}$ (8-way). This equaled an instance level $F_{\beta=1}$ of 96.7% (8-way). These were the second best scores in token-level evaluation and the best scores in instance-level evaluation among the systems submitted to the challenge, without any significant difference in performance from the other top-performing systems.

4.6 Comparison and conclusions

In the literature many de-identification approaches have been introduced. Some approaches target the recognition (and removal) of particular types of PHI like Taira et al.'s [64] system which focuses on patient names, or Thomas et al.'s method [65], which seeks to identify person names (both patients and doctors). There are several approaches that carry out the full de-identification of medical texts. These are based either on a pattern-matching algorithm that uses a thesaurus (Sweeny, [66], Ruch et al. [67]); a combination of rule-based systems and pattern matching using dictionaries (Douglass et al., [68]) and the Unified Medical Language System (Gupta et al., [69]), or on a statistical model (Sibanda and Uzuner, [70]).

The participants of the first Workshop on Challenges in Natural Language Processing for Clinical Data submitted both rule-based (Guillen [71]) and statistical approaches for the de-identification task. The best performing systems used Conditional Random Fields (Aramaki et al., [72]; Wellner at al. [73]); boosting and C4.5 decision tree learning (presented here) and Support Vector Machines (Hara [74]; Guo et al. [75]) to handle the anonymisation problem.

Our model achieved state-of-the-art accuracy and this shows the feasibility of adaptating our NER system, designed for newswire texts, to a biomedical free text processing task. We would like to emphasise here again that we achieved this competitive result without any deep knowledge information (not even POS codes) and without any domain specific resources. Our success is probably due to the very rich surface level and contextual feature representation.

The fact that our model – without any kind of fine tuning for the actual task – gives better results (99.48% 9-way token level accuracy) than the average performance score of the other submitted systems (99.19%) shows that the feature representation introduced in the previous sections is useful for NER in general. These kinds of features are simple and quick to implement, hence we think that our system can be used (or easily adapted) to other problems as well.

Similarly, the iterative learning approach seems to be a promising approach for documents that consists of parts with different characteristics (like discharge records having structured and unstructured parts).

As the systems participating in the challenge were trained and tested on a data set that contained re-identified PHI items, this forced them to rely entirely on contextual patterns, while some features that would undoubtedly help the recognition of real PHI (e.g. a list of possible first names) failed here. The artificially increased PHI/non-PHI ambiguity of the re-identified data made this task particularly challenging and the results on real-life data should be somewhat better in terms of recognising PHI items. On the other hand, inter-PHI ambiguity was moderate compared to real data thus cross-class labeling errors would probably occur more frequently than in a re-identified corpus. These types of errors are less serious though, as these do not lead to patient details being revealed.

4.7 Summary of Thesis results

The main results of this chapter can be summarised as follows.

Together with his collegues, the author participated in the 2006 I2B2 shared task challenge on medical record de-identification. The major steps of the adaptation of the pre-existing NER system, and results achieved (as a whole) are the joint contribution of the co-authors. As our results show, the system we obtained via the domain adaptation of our newswire NER model is competitive with other approaches, which means that our architecture is capable of solving NER tasks language and domain independently, with minimal adaptation effort.

In particular, the author made major contributions to the customisation of the feature representation, i.e. the development of novel features specifically for the medical domain. These novel features were helpful in achieving a state-of-the-art performance (our model had the best phrase-level 8-way F measure and second best token-level 9-way accuracy). The main beneficial characteristics of the novel features developed by us were the followings:

- Regular expressions enabled us to use complex, very powerful features to learn the recognition of well-formed PHI types.

- Structure information and iteratively generated features enabled us to thoroughly exploit the information hidden in the structure of records and the potentials in label-consistency.

Part II

Text Classification

Chapter 5

Smoker status classification in medical records

The principal reason for processing medical discharge records is to facilitate medical research carried out by physicians by providing them with statistically relevant data for analysis. An example of such an analysis might be a comparison of the runoff and effects of certain illnesses/diseases among patients with different social habits. The evidence drawn from the direct connection between social characteristics and diseases (like the link between smoking status and lung cancer or asthma) is of key importance in treatment and prevention issues. Such points can be deduced automatically by applying statistical methods on large corpora of medical records.

5.1 The smoking status identification task

Here we follow the task definition of the smoker challenge of the first I2B2 workshop. The task here was to classify medical records into the following five semantic classes based on the smoking status of the patient being examined:

- **non-smoker:** the patient has no smoking history,
  No tobacco.

- **current smoker:** he/she is an active smoker,
  She quit smoking four months ago.

- **past smoker:** the patient had not smoked for at least one year
  She is a past smoker, but quit two years ago when she was found to have right upper lobe nodule, which was resected and found to be positive for TB granuloma, for which she was treated with antibiotics for nine months.

- **smoker:** when the document contains no information about his current or past smoker status, but he/she has a smoking history,
  Depression, anxiety, chronic obstructive pulmonary disease/asthma, history of

tobacco abuse, chronic headaches, atypical chest pain with 6/97 Dobutamine MIBI revealing no ischemia and a history of tuberculosis exposure.

- **unknown:** the report contains no information about the patient's smoking status.
  Most recently, she developed dyspnea two days prior to admission, trigger was felt to be marijuana smoke in the building where she lives where there are many drug dealers.

5.1.1 Keyword-level classification

After some preliminary examination of the structure of medical discharge records, we came to the conclusion that it was not an entire discharge record that was required to extract the semantic information we desired, but just short excerpts from it contained sufficient information for us to identify patients belonging to different smoker classes. The absence of such excerpts on the other hand meant that the document contains no information on patient smoking status and thus it is labeled as unknown.

As the classification of smaller pieces of texts with the same information content is always easier to handle than bigger ones, we searched each document for relevant parts or sentences which frequently appeared in documents that belonged to one of the four smoker classes (referred to as *known* texts later on), but which were almost never seen in records that contained no information on the patient's smoking status.

The typical word stems that allowed us to identify *unknown* texts from others are listed in Table 5.1. To evaluate each keyword (or prefix), we used a feature ranking method that took into account the class-conditional probability of the keyword for *known* documents (predictive power) and the document frequency of the keyword (coverage). To avoid prioritising frequent words like stopwords we assigned very high weight to predictive power ($P(+|keyword)$ was powered by 10).

These word prefixes that were top ranked by our feature ranking method really tell us that a document contains relevant information on the smoking status of the patients. The most informative word chunks came to be {*ciga (for cigar & cigarette), smok (for smoke, smoking, ...), toba (for tobacco), habi (for habit), alco (for alcohol), fath (for father) and soci (for social)*}, which is an interesting but not surprising result. Since *Habit :*, *Social habit :* and *Social* are names of a header in discharge records and a heading usually contains sentences with one or more of the 3 other key words, we discarded these from our experiments. The word *alcohol* was indicative of *known* documents for the very same reason, while the high ranking of 'father' was a somewhat interesting result. It seems that physicians tend to include a description of the family history of the patient if he/she has social habits like smoking. As these above-mentioned keywords were all dependent on the remaining three, i.e. each time they appeared in the text, one of {*cigar, smoke or tobacco*} was also present, hence we restricted our classification schema to sentences containing any of these and discarded the rest of the top-ranked keywords. Actually there was one further keyword, *nicotine*, which is semantically similar to those we selected, but this one was underranked due to its

prefix	df(+)	df(-)	$P(+\|prefix)$	$P(+\|prefix)^{10} * df(+)/num(+)$
				feature weight
smok	93	2	97.89%	51.85%
toba	53	2	96.36%	25.24%
alco	79	8	90.80%	20.77%
soci	102	20	83.61%	11.74%
ciga	17	0	100.00%	11.72%
habi	18	1	94.74%	7.23%
fath	23	2	92.00%	6.89%
drin	43	7	86.00%	6.56%
etha	9	0	100.00%	6.21%
cold	7	0	100.00%	4.83%
obje	6	0	100.00%	4.14%
ca-1	6	0	100.00%	4.14%
bear	6	0	100.00%	4.14%
apar	6	0	100.00%	4.14%
aske	6	0	100.00%	4.14%
1250	6	0	100.00%	4.14%
lary	6	0	100.00%	4.14%
hers	5	0	100.00%	3.45%
97.6	5	0	100.00%	3.45%
thos	5	0	100.00%	3.45%
52-y	5	0	100.00%	3.45%
napr	5	0	100.00%	3.45%
illi	5	0	100.00%	3.45%
nico	5	0	100.00%	3.45%
opin	5	0	100.00%	3.45%
1984	5	0	100.00%	3.45%

Table 5.1: Document frequencies, class-conditional probabilities for the top ranked word-prefixes

relative rarity in the training corpus.

This way we built a keyword-level classifier, and because a document could contain more than one keyword, a joint decision had to be made to get a document-level classification. We used a simple majority voting rule to aggregate the predictions for each keyword to a single decision. We note here that the above-mentioned voting scheme was chosen arbitrarily and it is possible that other voting rules would have been more useful.

5.1.2 Description of our classification model

The basic steps of processing a discharge record are the following:

1. Preprocessing. This automatically filters out documents belonging to the un-

known class, and then collects relevant sentences from known-class documents.

2. The feature extractor builds a feature vector for each keyword found in the text for an inductive learning task (this step is described in detail in the next subsection).
3. A classifier model assigns one of the known-class labels (current smoker, non-smoker, past smoker, smoker) to each instance generated from the same document.
4. A majority voting scheme makes a final decision on which class the document belongs to.

5.1.3 Features used

Our smoker status classifier system uses similar features to those employed by Zeng et. al. [25], it considers phrases of length 1-3 words that we found characteristic to one or more of the smoker classes. In addition, we also tried to incorporate deeper knowledge about the meaning of the sentence with several helpful features by describing part of speech information or some very basic properties of the syntactic structure. To get POS and syntactic information we used the publicly available Link Parser [76]. We should mention here that the sentences we extracted from the discharge records were out-domain texts for the parser and were often poorly formed sentences. These facts made the results of the parser somewhat poorer in quality than expected, but we think that these sentences have very similar characteristics. Even the parse errors are similar in many cases and these features prove useful for the task.

The features we eventually opted for were the following:

1. We assigned 11 different values to the important 2-3 word long phrases for the class (or subset of classes) they indicated.
2. Which of the three keywords the sentence corresponded to.
3. Part of speech code of the keyword.
4. Whether the keyword was inside a Noun Phrase or Verb Phrase structure or not in the syntax tree of the sentence.
5. The lemma of the verb nearest to the keyword (in the syntax tree).
6. The part of speech code of the verb nearest to the keyword (in the syntax tree).
7. Whether the sentence contained a negative word (*no, none, never, negative, neither*) or not.
8. Words seen in the training data quite often (unigrams).

As regards the features described above, we collected 62 different attributes for each keyword in each sentence acquired from a document. The final decision on the patient's smoking status was made based on all the instances that originated from the same discharge summary, using a majority voting rule.

5.2 Feature selection

A high-dimensional feature space and limited amount of training data often leads to overfitting. In our case we had to handle the problem of having extremely low amounts of training data (about 200 instances) and numerous features collected for each instance. A common solution to avoid overfitting on the training data is to reduce the dimensionality of the feature space by means of feature selection, i.e. to discard irrelevant attributes and keep those few that have the highest predictive power.

χ^2 statistic (CSS): We used the well known χ^2 statistic to estimate the conditional dependence between individual features and the target attribute (that is, the class label). This statistical method computes the strength of dependency by comparing the joint distribution and the marginal distributions of the feature in question and the target variable. This way, the attributes could be ranked based on their individual relevance and this enabled us to discard insignificant features automatically.

CSS has some limitations though: e.g. it compares attributes to the target attribute just one at a time. Thus it is conceivable that when a feature is not really informative on its own, but is useful when combined with other attributes, it might get a low rank via the chi-squared statistic.

Best subset selection (BSS): Another possibility is to rank subsets of features together, rather than measuring their individual association with the class values. This method has a very high computational time complexity as the number of possible subsets of features grows exponentially with the dimensionality of the initial feature space. Since we had a rather low amount of training data available, this kind of subset evaluation became computationally feasible with classifiers that were fast to train, hence we decided to perform a best subset evaluation for the various attributes we used.

Both the CSS and BSS evaluations benefited from our deep knowledge features describing the syntactic and morphological properties of text, and important phrases of length 2-3 that indicated a single class value were also chosen by both evaluations. Best subset evaluation retained several features that described phrases indicating more than one class and several characteristic unigrams, while CSS underranked phrases that indicated 2 or more classes (indeed, these features proved to be useful in combination with others and CSS was barely able to capture this fact) and thus kept more unigram features, a few of which were hard to interpret.

The results of our feature evaluation clearly show that deep knowledge features which describe the syntactic properties of the text contribute greatly to the identification of a patient's smoking status. The features selected by one or both of the methods were the following:

Both: *lemma and POS of the verb nearest to keyword; negative word in the sentence; 2-3 word long phrases indicating 'current smoker', 'past smoker', 'non-smoker', 'current/past smoker' or 'smoker/non-smoker'; unigram in the sentence: 'ago'*

CSS: *POS of keyword; unigram in the sentence: 'years', 'does', 'smoke', 'per', 'smoker', 'approximately'*

BSS: *lemma of keyword; 2-3 word long phrases indicating 'smoker/current smoker' or 'smoker/past smoker'; keyword inside Noun Phrase; unigram in the sentence: 'use',*

'drinks', 'quitting'
As the features chosen by BSS were much easier to interpret, in our experiments we decided to use the 16 features that performed the best in the best subset selection process.

5.3 Results

Using the feature space representation described above, we trained several classifiers to predict the smoking status of the patients. All classifiers tested (Artificial Neural Network, Support Vector Machine, C4.5 decision tree, AdaBoostM1+C4.5) performed similarly, which shows that our features resulted in simple learnt patterns that any classifier could capture. Our models showed no statistically significant difference from each other, or any of the best submitted systems, which proves the feasibility of our excerpt-based (keyword level) classification model. We should say here though that the system of Clark et al. [77] was actually significantly better than any other participating systems, but their submissions suffered from some system errors. Their out-of-competition results, however were indeed significantly better than any of the other submitted systems, which shows that their model captured the information hidden in discharge summaries more accurately. They also prepared some additional training data and reported a boost in system performance, which demonstrates that systems trained on the challenge dataset suffered from a lack of training data, and could perform better if more examples were available to describe possible ways of expressing smoking status in narrative free texts.

5.3.1 Performance on the training set, applying 5-fold cross-validation

In Table 5.2 the document-level accuracies on the four known classes (discarding *unknown* documents) and for all five classes (including *unknown* documents) are given for all classifiers.

	4-class	5-class
k-NN	76.92	90.95
SVM	77.62	91.21
AB-C4.5	81.11	92.46
ANN	81.11	92.46
VOTE	83.22	93.22

Table 5.2: The document accuracy scores of our models

5.3.2 Performance on the i2b2 evaluation set

The behaviour of our best model was similar to the 5-fold result on the official i2b2 evaluation set (See Table 5.3). Our best model (the one using boosting and C4.5 decision trees) achieved a classification accuracy of 86.54% in 5-class evaluation, while the best performing system using the same data set had an accuracy of 88.79% [78]. One participant incorporated a significantly larger database for training purposes (over 1000 examples) and significantly outperformed all the other systems [77]. This clearly shows that the lack of training data is detrimental to the performance score.

	5-class	4-class	2-class
accuracy	86.54%	65.85%	90.24%

Table 5.3: The $F_{\beta=1}$ results based on the evaluation set. Here 5-class evaluation shows system performance when we distinguish between all five classes. 4-class evaluation excludes *unknown* documents and measures performance on the four known document classes, while 2-class evaluation differentiates between non-smoker and smoker classes (i.e. past- current- and smoker together).

	Un	Non	Pa	Sm	Cu
Unknown	63	0	0	0	0
Non-smoker	0	16	0	0	0
Past-smoker	0	1	5	1	4
Smoker	0	1	0	0	2
Current-smoker	0	2	3	0	6

Table 5.4: The confusion matrix of the AB+C4.5 model on the evaluation set

The confusion matrix for the i2b2 evaluation set is given in Table 5.4. We got significantly better results in 2-class evaluations (where we distinguish between patients with a smoking history and non-smokers, without dividing smoking patients into further subcategories), which surely demonstrates that the most challenging task for our classification model is separating current smokers, past smokers and smokers.

The two main reasons why the distinction between these three smoking classes proved to be the most difficult are probably the following. First, we had significantly fewer training examples for these three categories, and those patients that had quit smoking in the past year were treated as current smokers as their physiological characteristics were similar to current smokers. This way we also had to find out when they gave up smoking. Finally, reference to the time period when the patient's social habits changed were many times mentioned in separate sentences. If those sentences did not contain any keyword (only the preceding sentence, for example), we failed to extract this knowledge from the text. This is one of the most obvious limits of our model, and the problem needs to be handled somehow.

The i2b2 evaluation set used to rank the participating systems contained several cases where, considering the excerpts we collected on their own, the response of our model seemed more appropriate than the gold standard labeling. These cases are probably good examples for highlighting the limitations of our approach as the physicians must have found evidence elsewhere in the document to support their judgement. It seems our system was unable to locate this additional place of information. An example of such an excerpt is:

No alcohol use and quit tobacco greater than 25 years ago with a 10-pack year (our system: PAST/ gold standard label: CURRENT)

All the other system errors we encountered were the kind of cases where a human expert was able to make the proper decision based on the limited data we extracted from the whole discharge record. In the future it would be good to eliminate these errors as best one can.

We should also mention here that those cases where the annotators apparently found additional information in the documents were typically elements of one of three smoker classes. This shows that making a distinction between smokers is more difficult for experts as well. Also, their rate of agreement is lower on the smoker classes than on non-smoker and unknown documents.

5.4 Comparison and conclusions

The identification of smoking habits based on discharge records was studied earlier in the literature. [25; 79] reported an accuracy of 90% on the identification of smoker status. They constructed a classification model using about 8500 smoking-related sentences obtained from discharge records and the Support Vector Machine (SVM) as a classifier with word phrases of length 1-3 as features. Our approach differs from the one reported by them in the amount of data used (about 200 smoking-related sentences) and the variety of features employed (our system exploits syntactic information as well).

A detailed comparison of the systems that were entered the challenge can be found in [26]. Among these were both manually constructed systems and machine learning approaches. Pedersen [80] applied supervised (decision tree) and unsupervised models (clustering) to the task using simple vector space representation of the documents. Aramaki et al.[78] applied a two-step classifier (like the one described above) and first extracted relevant sentences for the smoking status of patients. If multiple sentences contained smoker cues, they used the last sentence in the document though. Using the sentences they extracted they applied Okapi-BM25 and k-nearest-neighbors to determine the proper label of the document (one of the four known classes). Carrero et al. [81] tested several machine learning methods (those used here and Naïve Bayes classifier) using, uni-, bi- and trigram features. Similarly to our results, they found bigrams and trigrams to be more useful for smoker classification than unigrams. Clark et al. [77] used SVMs and MaxEnt classifiers for the task that benefited from document structure information, medical entity tagging (including the recognition of smoking-related medication, like *Nicoderm*), and also tested the results of incorporating additional training data to the system. Their results indicated that both the use of a Medical Extraction system and larger annotated dataset benefits performance. Regarding their classifiers

	Macroaveraged			Microaveraged		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Clark_3	0.81	0.73	0.76	0.90	0.90	0.90
Cohen_2	0.64	0.67	0.65	0.88	0.89	0.89
Aramaki_1	0.64	0.67	0.65	0.88	0.89	0.88
Cohen_1	0.64	0.65	0.64	0.88	0.88	0.88
Clark_2	0.76	0.69	0.72	0.87	0.88	0.88
Cohen_3	0.62	0.62	0.62	0.87	0.88	0.87
Wicentowski_1	0.58	0.61	0.59	0.85	0.87	0.86
Szarvas_2	0.59	0.60	0.59	0.85	0.87	0.85
Clark_1	0.69	0.65	0.66	0.86	0.87	0.85
Szarvas_3	0.56	0.58	0.57	0.84	0.86	0.84
Savova_1	0.62	0.60	0.60	0.84	0.86	0.84
Szarvas_1	0.56	0.58	0.57	0.83	0.86	0.84
Sheffer_1	0.59	0.59	0.58	0.83	0.86	0.84
Savova_2	0.56	0.57	0.56	0.81	0.84	0.82
Savova_3	0.55	0.55	0.55	0.80	0.83	0.81
Pedersen_1	0.55	0.56	0.54	0.82	0.82	0.81
Guillen_1	0.45	0.51	0.44	0.77	0.79	0.76
Carrero_1	0.52	0.47	0.48	0.74	0.77	0.75
Carrero_2	0.44	0.43	0.41	0.71	0.71	0.70
Rekdal_1	0.68	0.45	0.47	0.77	0.74	0.67
Pedersen_3	0.23	0.35	0.27	0.53	0.68	0.60
Pedersen_2	0.23	0.36	0.28	0.53	0.69	0.59
Carrero_3	0.26	0.31	0.27	0.54	0.63	0.57

Table 5.5: Microaverages and Macroaverages for Precision, Recall, and F-measure, sorted by microaveraged F-measure

they experimented with sentence and phrase-level approaches and incorporated multiple decisions using a majority voting rule similar to ours. Cohen [82] defined the scope for processing as the ±100 characters context of a smoker cue and applied a linear SVM to solve the classification task. Savova et al. [83] applied a three-stage approach that implemented a hierarchical classification using SVM models. They also filtered out documents with no smoker cues. In a three-stage approach they first attempted to classify a non-smoker label (this model benefited negation detection) and then tried to differentiate between past smoker and current smoker classes (this model used non-lemmatised texts). The final document-level decision was the label of highest priority that has been assigned to the document (their priority order was current smoker, past smoker, smoker, non- smoker, and unknown). Sheffer et al.[84] adapted for the smoker task and applied an expert system designed to process medical texts.

A detailed comparison of the above-mentioned systems is given in Table 5.5[1].

Observing that most systems attained a remarkable accuracy using the very limited

[1] We note here that the submissions of Clark et al.[77] suffered from some implementation errors and their real F scores are slightly higher than those listed here. This table is taken from [26]

amount of training data provided, we conclude that this task can be accomplished with good accuracy and these systems can be useful in practice.

5.5 Summary of thesis results

The main results of this chapter can be summarised as follows.

Together with his collegues, the author participated in the 2006 I2B2 shared task challenge on patient smoking status classification from medical records. The system and the overall results we submitted are a shared and indivisible contribution of the co-authors.

In particular, the author made major contributions to the design of the feature representation, i.e. the development of features used by previous studies and novel ones specifically for the medical domain which tried to group more or less similar examples together by exploiting the syntactic or semantic classification of phrases. The main reasoning for having these novel features was to reduce the effects of a small sample size. These novel features were helpful in achieving a good performance (they appeared among the top ranked attributes using 2 different feature selection methods). The main beneficial characteristics of our approach were the following:

- Using class conditional probability measures, we succesfully extracted those few keywords that covered all textual appearance of smoking status information and allowed us to separate unknown documents with 100% accuracy.
- Our novel features and keyword level model proved to be useful for the smoker status classification task even with very limited training data available. With a bigger dataset we expect the overall performance score to be even higher.

Chapter 6

Identifying speculations in biomedical texts

The highly accurate identification of several regularly occurring language phenomena like the speculative use of language, negation and past tense (temporal resolution) is a prerequisite for the efficient processing of biomedical texts. In various natural language processing tasks, relevant statements appearing in a speculative context are treated as false positives. Hedge detection seeks to perform a kind of semantic filtering of texts, that is it tries to separate factual statements from speculative/uncertain ones.

6.1 Hedging in biomedical NLP

To demonstrate the detrimental effects of speculative language on biomedical NLP tasks, we will consider two inherently different sample tasks, namely the ICD-9-CM coding of radiology records and gene information extraction from biomedical scientific texts. The general features of texts used in these tasks differ significantly from each other, but both tasks require the exclusion of uncertain (or speculative) items from processing.

6.1.1 Gene Name and interaction extraction from scientific texts

In our experiments we used the dataset made available by Medlock and Briscoe [29]. The dataset consists of a training set generated semi-automatically (this process will be described later on) and a manually annotated test set of five full papers from FlyBase. The main characteristics of the manually annotated test set are shown in Table 6.1.

The test set of the hedge classification dataset [1] [29] has also been annotated for gene names[2].

[1] http://www.cl.cam.ac.uk/~bwm23/

[2] http://www.cl.cam.ac.uk/~nk304/

Articles	5
Sentences	1537
Spec sentences	380
Nspec sentences	1157

Table 6.1: Characteristics of the FlyBase hedge dataset.

Examples of speculative assertions:

Thus, the D-mib wing phenotype may result from defective N inductive signaling at the D-V boundary.

A similar role of Croquemort has not yet been tested, but seems likely since the crq mutant used in this study (crqKG01679) is lethal in pupae.

After an automatic parallelisation of the 2 annotations (sentence matching) we found that a significant part of the gene names mentioned (638 occurences out of a total of 1968) appears in a speculative sentence. This means that approximately 1 in every 3 genes should be excluded from the interaction detection process. These results suggest that a major portion of system false positives could be due to hedging if hedge detection had been neglected by a gene interaction extraction system.

	spec	nspec
spec	489	118
nspec	25	1171

Table 6.2: The confusion matrix for gene names mentioned in a speculative context (rows – *spec*, *nspec* instances; cols – classified as *spec*, *nspec*).

The first row lists the number of speculative instances, while the second row shows the number of non-speculative gene name mentions. The first column shows instances classified as speculative, while the second column shows instances classified as non-speculative.

Table 6.2 shows the confusion matrix of our system for classifying gene name mentions based on their appearance in the text (in a speculative or non-speculative context). We consider this a useful result as such an accurate preprocessing step would surely boost the performance of end-user applications like gene interaction extraction systems.

Following the annotation standards of Medlock and Briscoe [29], we manually annotated 4 full articles downloaded from the BMC Bioinformatics website to evaluate our final model on documents from an external source. The chief characteristics of this dataset (which is available at[3]) is shown in Table 6.3.

[3] http://www.inf.u-szeged.hu/~szarvas/homepage/hedge.html

Articles	4
Sentences	1087
Spec sentences	190
Nspec sentences	897

Table 6.3: Characteristics of the BMC hedge dataset.

6.1.2 ICD-9-CM coding of radiology records

Automating the assignment of ICD-9-CM codes for radiology records was the subject of a shared task challenge organised in the spring of 2007. The detailed description of the task, and the challenge itself can be found in [31] and online[4]. ICD-9-CM codes that are assigned to each report after the patient's clinical treatment are used for the reimbursement process by insurance companies. There are official guidelines for coding radiology reports [85]. These guidelines strictly state that an uncertain diagnosis should never be coded, hence identifying reports with a diagnosis in a speculative context is an inevitable step in the development of automated ICD-9-CM coding systems. The following examples illustrate a typical non-speculative context where a given code should be added, and a speculative context where the same code should never be assigned to the report:

non-speculative: *Subsegmental **atelectasis** in the left lower lobe, otherwise normal exam.*

speculative: *Findings suggesting viral or reactive airway disease with right lower lobe **atelectasis** or pneumonia.*

In an ICD-9 coding system developed for the challenge, the inclusion of a hedge classifier module (a simple keyword-based lookup method with 38 keywords) improved the overall system performance from 79.7% to 89.3%.

To evaluate our system we annotated the ICD-9-CM coding dataset for hedging. We considered each sentence that contained a disease or symptom name recognised by our ICD-9-CM coding application for hedge annotation – if a sentence contained some speculative element is was classified as speculative and non-speculative otherwise. Sentences that did not contain any medical terminology were discarded from the dataset.

Sentences	3951
Spec sentences	633
Nspec sentences	3318

Table 6.4: Characteristics of the BMC hedge dataset.

[4] http://www.computationalmedicine.org/challenge/index.php

6.2 Description of our system and the training data generation methods

6.2.1 Feature space representation

Hedge classification can essentially be handled by acquiring task specific keywords that trigger speculative assertions more or less independently of each other. As regards the nature of this task, a vector space model (VSM) is a straightforward and suitable representation for statistical learning. As VSM is inadequate for capturing the (possibly relevant) relations between subsequent tokens, we decided to extend the representation with bi- and trigrams of words. We chose not to add any weighting of features (by frequency or importance) and for the Maximum Entropy Model classifier we included binary data about whether single features occurred in the given context or not.

6.2.2 Training data acquisition

Probabilistic approach

To build our classifier models, we used the dataset gathered and made available by [29]. They commenced with the seed set S_{spec} gathered automatically (all sentences containing *suggest* or *likely* – two very good speculative keywords), and S_{nspec} that consisted of randomly selected sentences from which the most probable speculative instances were filtered out by a pattern matching and manual supervision procedure. With these seed sets they then performed the following iterative method to enlarge the initial training sets, adding examples to both classes from an unlabeled pool of sentences called U:

1. Generate seed training data: S_{spec} and S_{nspec}
2. Initialise: $T_{spec} \leftarrow S_{spec}$ and $T_{nspec} \leftarrow S_{nspec}$
3. Iterate:
    - Train classifier using T_{spec} and T_{nspec}
    - Order U by $P(spec)$ values assigned by the classifier
    - $T_{spec} \leftarrow$ most probable batch
    - $T_{nspec} \leftarrow$ least probable batch

What makes this iterative method efficient is that, as we said earlier, hedging is expressed via keywords in natural language texts; and often several keywords are present in a single sentence. The seed set S_{spec} contained either *suggest* or *likely*, and due to the fact that other keywords cooccur with these two in many sentences, they appeared in S_{spec} with reasonable frequency. As sentences with cooccuring words were found more likely to be speculative than those containing no cooccuring words at all, these instances were selected for manual filtering. Thus S_{nspec} was filtered from the most common hedge cues just by examining a small portion of the set manually. This way the initial seed sets contained some good indicators of speculative use besides *suggest* and *likely*. For example, $P(spec|may) = 0.9985$ on the seed sets created by [29]. The iterative extension of the training sets for each class further boosted this effect, and skewed the distribution of speculative indicators as sentences containing them were likely to be added to the extended training set for the speculative class, and unlikely to fall into the non-speculative set.

We should add here that the very same feature has an inevitable, but very important side effect that is detrimental to the classification accuracy of models trained on a dataset which has been obtained this way. This side effect is that other words (often common words or stopwords) that tend to cooccur with hedge cues will also be subject to the same iterative distortion of their distribution in speculative and non-speculative uses. Perhaps the best example of this is the word *it*. Being a stopword in our case, and having no relevance at all to speculative assertions, it has a class conditional probability of $P(spec|it) = 74.67\%$ on the seed sets. This is due to the use of phrases like *it suggests that*, *it is likely*, and so on. After the iterative extension of training sets, the class-conditional probability of *it* dramatically increased, to $P(spec|it) = 94.32\%$. This is a consequence of the frequent co-occurence of *it* with meaningful hedge cues and the probabilistic model used and happens with many other irrelevant terms (not just stopwords). The automatic elimination of these irrelevant candidates is one of our main goals (to limit the number of candidates for manual consideration and thus to reduce the human effort required to select meaningful hedge cues).

This shows that, in addition to the desired effect of introducing further speculative keywords and biasing their distribution towards the speculative class, this iterative process also introduces significant noise into the dataset. This observation led us to the conclusion that in order to build efficient classifiers based on this kind of dataset, we should filter out noise. In the next part we will present our feature selection procedure (evaluated in the Results section) which is capable of underranking irrelevant keywords in the majority of cases.

Automatic training data generation

We present results for the automatic detection of speculative assertions in radiology reports. Here we generated training data by an automated procedure. Since hedge cues cause systems to predict false positive labels, our idea here was to train Maximum Entropy Models for the false positive classifications of our ICD-9-CM coding system using the vector space representation of radiology reports. That is, we classified every sentence that contained a medical term (disease or symptom name) and caused the automated ICD-9 coder[5] to predict a false positive code was treated as a speculative sentence and all the rest were treated as non-speculative sentences.

Here a significant part of the false positive predictions of the ICD-9-CM coding

[5] Here the ICD-9 coding system did not handle the hedging task.

system that did not handle hedging originated from speculative assertions, which led us to expect that we would have the most hedge cues among the top ranked keywords which implied false positive labels.

6.2.3 Feature (or keyword) selection

To handle the inherent noise in the training dataset that originates from its weakly supervised construction, we applied the following feature selection procedure. The main idea behind it is that it is unlikely that more than two keywords are present in the text, which are useful for deciding whether an instance is speculative. Here we performed the following steps:

1. We ranked the features x by frequency and their class conditional probability $P(spec|x)$. We then selected those features that had $P(spec|x) > 0.94$ (this threshold was chosen arbitrarily) and appeared in the training dataset with reasonable frequency (frequency above 10^{-5}). This set constituted the 2407 candidates which we used in the second analysis phase.

2. For trigrams, bigrams and unigrams – processed separately – we calculated a new class-conditional probability for each feature x, discarding those observations of x in speculative instances where x was not among the two highest ranked candidate. Negative credit was given for all occurrences in non-speculative contexts. We discarded any feature that became unreliable (i.e. any whose frequency dropped below the threshold or the strict class-conditional probability dropped below 0.94). We did this separately for the uni-, bi- and trigrams to avoid filtering out longer phrases because more frequent, shorter candidates took the credit for all their occurrences. In this step we filtered out 85% of all the keyword candidates and kept 362 uni-, bi-, and trigrams altogether.

3. In the next step we re-evaluated all 362 candidates together and filtered out all phrases that had a shorter and thus more frequent substring of themselves among the features, with a similar class-conditional probability on the speculative class (worse by 2% at most). Here we discarded a further 30% of the candidates and kept 253 uni-, bi-, and trigrams altogether.

This efficient way of reranking and selecting potentially relevant features (we managed to discard 89.5% of all the initial candidates automatically) made it easier for us to manually validate the remaining keywords. This allowed us to incorporate supervision into the learning model in the feature representation stage, but keep the weakly supervised modelling (with only 5 minutes of expert supervision required).

6.2.4 Maximum Entropy Classifier

Maximum Entropy Models [32] seek to maximise the conditional probability of classes, given certain observations (features). This is performed by weighting features to maximise the likelihood of data and, for each instance, decisions are made based on features present at that point, thus maxent classification is quite suitable for our purposes. As feature weights are mutually estimated, the maxent classifier is capable of taking feature dependence into account. This is useful in cases like the feature *it* being dependent on others when observed in a speculative context. By downweighting such features, maxent is capable of modelling to a certain extent the special characteristics which arise from the automatic or weakly supervised training data acquisition procedure. We used the OpenNLP maxent package, which is freely available[6].

6.3 Results

In this section we will present our results for hedge classification as a standalone task. In experiments we made use of the hedge classification dataset of scientific texts provided by [29] and used a labeled dataset generated automatically based on false positive predictions of an ICD-9-CM coding system.

6.3.1 Results for hedge classification in biomedical texts

As regards the degree of human intervention needed, our classification and feature selection model falls within the category of weakly supervised machine learning. In the following sections we will evaluate our above-mentioned contributions one by one, describing their effects on feature space size (efficiency in feature and noise filtering) and classification accuracy. In order to compare our results with Medlock and Briscoe's results [29], we will always give the $BEP(spec)$ that they used – the break-even-point of precision and recall[7]. We will also present $F_{\beta=1}(spec)$ values which show how good the models are at recognising speculative assertions.

The effects of automatic feature selection

The method we proposed seems especially effective in the sense that we successfully reduced the number of keyword candidates from an initial 2407 words having $P(spec|x) > 0.94$ to 253, which is a reduction of almost 90%. During the process, very few useful keywords were eliminated and this indicated that our feature selection procedure was capable of distinguishing useful keywords from noise (i.e. keywords having a very high speculative class-conditional probability due to the skewed characteristics of the automatically gathered training dataset). The 2407-keyword model achieved a $BEP(spec)$ os 76.05% and $F_{\beta=1}(spec)$ of 73.61%, while the model after feature selection performed better, achieving a $BEP(spec)$ score of 78.68% and $F_{\beta=1}(spec)$ score of 78.09%. Simplifying the model to predict a $spec$ label each time a keyword was present (by discarding those 29 features that were too weak to predict

[6] http://maxent.sourceforge.net/

[7] It is the point on the precision-recall curve of $spec$ class where $P = R$. If an exact $P = R$ cannot be realised due to the equal ranking of many instances, we use the point closest to $P = R$ and set $BEP(spec) = (P + R)/2$. BEP is an interesting metric as it demonstrates how well we can trade-off precision for recall.

$spec$ alone) slightly increased both the $BEP(spec)$ and $F_{\beta=1}(spec)$ values to 78.95% and 78.25%. This shows that the Maximum Entropy Model in this situation could not learn any meaningful hypothesis from the cooccurence of individually weak keywords.

Improvements by manual feature selection

After a dimension reduction via a strict reranking of features, the resulting number of keyword candidates allowed us to sort the retained phrases manually and discard clearly irrelevant ones. We judged a phrase irrelevant if we could consider no situation in which the phrase could be used to express hedging. Here 63 out of the 253 keywords retained by the automatic selection were found to be **potentially** relevant in hedge classification. All these features were sufficient for predicting the $spec$ class alone, thus we again found that the learnt model reduced to a single keyword-based decision.[8] These 63 keywords yielded a classifier with a $BEP(spec)$ score of 82.02% and $F_{\beta=1}(spec)$ of 80.88%.

Results obtained adding external dictionaries

In our final model we added the keywords used in [27] and those gathered for our ICD-9-CM hedge detection module. Here we decided not to check whether these keywords made sense in scientific texts or not, but instead left this task to the maximum entropy classifier, and added only those keywords that were found reliable enough to predict $spec$ label alone by the maxent model trained on the training dataset. These experiments confirmed that hedge cues are indeed task specific – several cues that were reliable in radiology reports proved to be of no use for scientific texts. We managed to increase the number of our features from 63 to 71 using these two external dictionaries.

These additional keywords helped us to increase the overall coverage of the model. Our final hedge classifier yielded a $BEP(spec)$ score of 85.29% and $F_{\beta=1}(spec)$ score of 85.08% (89.53% Precision, 81.05% Recall) for the speculative class. This meant an overall classification accuracy of 92.97%.

Using this system as a pre-processing module for a hypothetical gene interaction extraction system, we found that our classifier successfully excluded gene names mentioned in a speculative sentence (it removed 81.66% of all speculative mentions) and this filtering was performed with a respectable precision of 93.71% ($F_{\beta=1}(spec) = 87.27\%$).

Evaluation on scientific texts from a different source

Surprisingly, the model learnt on FlyBase articles seemed to generalise to these texts only to a limited extent. Our hedge classifier model yielded a $BEP(spec) = 75.88\%$ and $F_{\beta=1}(spec) = 74.93\%$ (mainly due to a drop in precision), which is unexpectedly low compared to the previous results.

[8] We kept the test set blind during the selection of relevant keywords. This meant that some of them eventually proved to be irrelevant, or even lowered the classification accuracy. Examples of such keywords were *will, these data* and *hypothesis*. We (falsely) assumed that these might suggest a speculative assertion.

Analysis of errors revealed that some keywords which proved to be very reliable hedge cues in FlyBase articles were also used in non-speculative contexts in the BMC articles. Over 50% (24 out of 47) of our false positive predictions were due to the different use of 2 keywords, *possible* and *likely*. These keywords were many times used in a mathematical context (referring to probabilities) and thus expressed no speculative meaning, while such uses were not represented in the FlyBase articles (otherwise bigram or trigram features could have captured these non-speculative uses).

The effect of using 2-3 word-long phrases as hedge cues

Our experiments demonstrated that it is indeed a good idea to include longer phrases in the vector space model representation of sentences. One third of the features used by our advanced model were either bigrams or trigrams. About half of these were the kind of phrases that had no unigram components of themselves in the feature set, so these could be regarded as meaningful standalone features. Examples of such speculative markers in the fruit fly dataset were: *results support, these observations, indicate that, not clear, does not appear, ...* The majority of these phrases were found to be reliable enough for our maximum entropy model to predict a speculative class based on that single feature.

Our model using just unigram features achieved a $BEP(spec)$ score of 78.68% and $F_{\beta=1}(spec)$ score of 80.23%, which means that using bigram and trigram hedge cues here significantly improved the performance (the difference in $BEP(spec)$ and $F_{\beta=1}(spec)$ scores were 5.23% and 4.97%, respectively).

6.3.2 Results for hedge classification in radiology reports

In this section we present results using the above-mentioned methods for the automatic detection of speculative assertions in radiology reports. Here we generated training data by an automated procedure. Since hedge cues cause systems to predict false positive labels, our idea here was to train Maximum Entropy Models for the false positive classifications of our ICD-9-CM coding system using the vector space representation of radiology reports. That is, we classified every sentence that contained a medical term (disease or symptom name) and caused the automated ICD-9 coder[9] to predict a false positive code was treated as a speculative sentence and all the rest were treated as non-speculative sentences.

Here a significant part of the false positive predictions of an ICD-9-CM coding system that did not handle hedging originated from speculative assertions, which led us to expect that we would have the most hedge cues among the top ranked keywords which implied false positive labels.

Taking the above points into account, we used the training set of the publicly available ICD-9-CM dataset to build our model and then evaluated each single token by this model to measure their predictivity for a false positive code. Not surprisingly,

[9] Here the ICD-9 coding system did not handle the hedging task.

some of the best hedge cues appeared among the highest ranked features, while some did not (they did not occur frequently enough in the training data to be captured by statistical methods).

For this task, we set the initial $P(spec|x)$ threshold for filtering to 0.7 since the dataset was generated by a different process and we expected hedge cues to have lower class-conditional probabilities without the effect of the probabilistic data acquisition method that had been applied for scientific texts. Using all 167 terms as keywords that had $P(spec|x) > 0.7$ resulted in a hedge classifier with an $F_{\beta=1}(spec)$ score of 64.04%.

After the feature selection process 54 keywords were retained. This 54-keyword maxent classifier got an $F_{\beta=1}(spec)$ score of 79.73%. Plugging this model (without manual filtering) into the ICD-9 coding system as a hedge module, the ICD-9 coder yielded an F measure of 88.64%, which is much better than one without a hedge module (79.7%).

Our experiments revealed that in radiology reports, which mainly concentrate on listing the identified diseases and symptoms (facts) and the physician's impressions (speculative parts), detecting hedge instances can be performed accurately using unigram features. All bi- and trigrams retained by our feature selection process had unigram equivalents that were eliminated due to the noise present in the automatically generated training data.

We manually examined all keywords that had a $P(spec) > 0.5$ given as a standalone instance for our maxent model, and constructed a dictionary of hedge cues from the promising candidates. Here we judged 34 out of 54 candidates to be potentially useful for hedging. Using these 34 keywords we got an $F_{\beta=1}(spec)$ performance of 81.96% due to the improved precision score.

Extending the dictionary with the keywords we gathered from the fruit fly dataset increased the $F_{\beta=1}(spec)$ score to 82.07% with only one out-domain keyword accepted by the maxent classifier.

6.3.3 Summary of results

The overall results of our study are summarised in a concise way in Table 6.5. We list $BEP(spec)$ and $F_{\beta=1}(spec)$ values for the scientific text dataset, and $F_{\beta=1}(spec)$ for the clinical free text dataset. Baseline 1 denotes the substring matching system of Light et al. [27] and Baseline 2 denotes the system of Medlock and Briscoe [29]. For clinical free texts, Baseline 1 is an out-domain model since the keywords were collected for scientific texts by [27]. The third row corresponds to a model using all keywords $P(spec|x)$ above the threshold and the fourth row a model after automatic noise filtering, while the fifth row shows the performance after the manual filtering of automatically selected keywords. The last row shows the benefit gained by adding reliable keywords from an external hedge keyword dictionary.

	Biomedical papers		Medical reports
	$BEP(spec)$	$F_{\beta=1}(spec)$	$F_{\beta=1}(spec)$
Baseline 1	60.00	–	48.99
Baseline 2	76.30	–	–
All features	76.05	73.61	64.04
Feature selection	78.68	78.09	79.73
Manual feat. sel.	82.02	80.88	81.96
Outer dictionary	85.29	85.08	82.07

Table 6.5: Summary of results.

6.4 Comparison and conclusions

6.4.1 Related work

Although a fair amount of literature on hedging in scientific texts has been produced since the 1990s (e.g. [28]), speculative language from a Natural Language Processing perspective has only been studied in the past few years. This phenomenon, together with others used to express forms of authorial opinion, is often classified under the notion of subjectivity [86], [23]. Previous studies [27] showed that the detection of hedging can be solved effectively by looking for specific keywords which imply that the content of a sentence is speculative and constructing simple expert rules that describe the circumstances of where and how a keyword should appear. Another possibility is to treat the problem as a classification task and train a statistical model to discriminate speculative and non-speculative assertions. This approach requires the availability of labeled instances to train the models on. Riloff et al. [87] applied bootstrapping to recognise subjective noun keywords and classify sentences as subjective or objective in newswire texts. Medlock and Briscoe [29] proposed a weakly supervised setting for hedge classification in scientific texts where the aim is to minimise human supervision needed to obtain an adequate amount of training data.

Here we followed [29] and treat the identification of speculative language as the classification of sentences for either speculative or non-speculative assertions, and extend their methodology in several ways. Thus given labeled sets S_{spec} and S_{nspec} we trained a model that, for each sentence s, is capable of deciding whether a previously unseen s contains a speculative element or not.

6.4.2 Conclusions

Our results presented above confirm our hypothesis that speculative language plays an important role in the biomedical domain, and it should be handled in various NLP applications. We experimentally compared the general features of this task in texts from two different domains, namely medical free texts (radiology reports), and scientific articles on the fruit fly from FlyBase.

The radiology reports had mainly unambiguous single-term hedge cues. On the other hand, it proved to be useful to consider bi- and trigrams as hedge cues in scientific texts. This, and the fact that many hedge cues were found to be ambiguous (they appeared in both speculative and non-speculative assertions) can be attributed to the literary style of the articles. Next, as the learnt maximum entropy models show, the hedge classification task reduces to a lookup for single keywords or phrases and to the evaluation of the text based on the most relevant cue alone. Removing those features that were insufficient to classify an instance as a hedge individually did not produce any significant difference in the $F_{\beta=1}(spec)$ scores. This latter fact justified a view of ours, namely that during the construction of a statistical hedge detection module for a given application the main issue is to find the task-specific keywords.

Our findings based on the two datasets employed show that automatic or weakly supervised data acquisition, combined with automatic and manual feature selection to eliminate the skewed nature of the data obtained, is a good way of building hedge classifier modules with an acceptable performance.

The analysis of errors indicate that more complex features like dependency structure and clausal phrase information could only help in allocating the scope of hedge cues detected in a sentence, not the detection of any itself. Our finding that token unigram features are capable of solving the task accurately agrees with the the results of previous works on hedge classification ([27], [29]), and we argue that 2-3 word-long phrases also play an important role as hedge cues and as non-speculative uses of an otherwise speculative keyword as well (i.e. to resolve an ambiguity). In contrast to the findings of Wiebe et al. ([86]), who addressed the broader task of subjectivity learning and found that the density of other potentially subjective cues in the context benefits classification accuracy, we observed that the co-occurence of speculative cues in a sentence does not help in classifying a term as speculative or not. Realising that our learnt models never predicted speculative labels based on the presence of two or more individually weak cues and discarding such terms that were not reliable enough to predict a speculative label (using that term alone as a single feature) slightly improved performance, we came to the conclusion that even though speculative keywords tend to cooccur, and two keywords are present in many sentences; hedge cues have a speculative meaning (or not) on their own without the other term having much impact on this.

The main issue thus lies in the selection of keywords, for which we proposed a procedure that is capable of reducing the number of candidates to an acceptable level for human evaluation – even in data collected automatically and thus having some undesirable properties.

The worse results on biomedical scientific papers from a different source also corroborates our finding that hedge cues can be highly ambiguous. In our experiments two keywords that are practically never used in a non-speculative context in the FlyBase articles we used for training were responsible for 50% of false positives in BMC texts since they were used in a different meaning. In our case, the keywords *possible* and *likely* are apparently always used as speculative terms in the FlyBase articles used, while the articles from BMC Bioinformatics frequently used such cliche phrases as *all possible combinations* or *less likely / more likely . . .* (referring to probabilities shown in the figures). This shows that the portability of hedge classifiers is limited, and cannot really be done without the examination of the specific features of target texts or a more heterogenous corpus is required for training. The construction of hedge classifiers for each separate target application in a weakly supervised way seems feasible though. Collecting bi- and trigrams which cover non-speculative usages of otherwise common hedge cues is a promising solution for addressing the false positives in hedge classifiers and for improving the portability of hedge modules.

6.4.3 Resolving the scope of hedge keywords

In this paper we focused on the recognition of hedge cues in texts. Another important issue would be to determine the scope of hedge cues in order to locate uncertain sentence parts. This can be solved effectively using a parser adapted for biomedical papers. We manually evaluated the parse trees generated by [88] and came to the conclusion that for each keyword it is possible to define the scope of the keyword using subtrees linked to the keyword in the predicate-argument syntactic structure or by the immediate subsequent phrase (e.g. prepositional phrase). Naturally, parse errors result in (slightly) mislocated scopes but we had the general impression that state-of-the-art parsers could be used efficiently for this issue. On the other hand, this approach requires a human expert to define the scope for each keyword separately using the predicate-argument relations, or to determine keywords that act similarly and their scope can be located with the same rules. Another possibility is simply to define the scope to be each token up to the end of the sentence (and optionally to the previous punctuation mark). The latter solution has been implemented by us and works accurately for clinical free texts. This simple algorithm is similar to NegEx [89] as we use a list of phrases and their context, but we look for punctuation marks to determine the scopes of keywords instead of applying a fixed window size.

6.5 Summary of Thesis results

The major results of this thesis can be summarised as follows.

Here we revisited Medlock and Briscoe's [29] weakly supervised model for hedge classification. As the results show, accurate feature selection based on ranking (and iterative reranking) of token n-gram features can successfully extract typical hedge cues from datasets labeled by weakly or unsupervised methods. By means of automatic (and finally manual) feature selection, reasonable improvements in hedge classification accuracy can be achieved. We demonstrated our findings using the dataset introduced by Medlock and Briscoe's [29], and also using a different application domain, i.e. medical free texts. We experimentally compared the different use of hedging in these application scenarios, and using biological articles from a different source than those used by Medlock and Briscoe we also showed that the construction of hedge classifiers could not be carried out domain independently.

All the contributions in this chapter are independent results of the author. The major findings of this chapter are the following:

- The construction of a complex feature ranking and selection procedure that successfully reduces the number of keyword candidates (those having the highest class-conditional probability for hedge class) without excluding helpful hedge keywords.
- We demonstrated that with a very limited amount of expert supervision in finalising the feature representation, it is possible to build accurate hedge classifiers from semi-automatically or automatically collected training data.
- We extended the scope of evaluations to two applications with different kinds of texts involved (scientific articles used in previous works, and also medical free texts).
- The extension of the feature representation to 2-3 word-long phrases and an evaluation of the importance of longer keywords in hedge classification.
- We demonstrated (using a small test corpora of biomedical scientific papers from a different source) that hedge keywords are highly task-specific and thus constructing models that generalise well from one task to another is not feasible without a noticeable loss in accuracy.

Our findings also demonstrate that statistical models trained to classify hedge and non-hedge sentences are unable to learn complex inter-dependencies between hedge cues and simplify to classification based on a single feature (either a unigram or longer phrase) rather than on the combination of several hedge cues. This means that the main problem in hedge detection is to find the highly task-specific keywords.

Chapter 7

The automatic construction of rule-based ICD-9-CM coding systems

Clinical coding, i.e. the assignment of International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM) codes to medical documents serves as a justification for carrying out a certain procedure. This means that the reimbursement process by insurance companies is based on the labels that are assigned to each report after the patient's clinical treatment. The approximate cost of ICD-9-CM coding clinical records and correcting related errors is estimated to be about $25 billion per year in the US [30]. There are official guidelines for coding radiology reports [85]. These guidelines define the codes for each disease and symptom and also place restrictions on how and when certain codes can be applied. Such constraints include the following:

- an uncertain diagnosis should never be coded,
- symptoms should be omitted when a certain diagnosis that is connected with the symptom in question is present and
- past illnesses or treatments that have no direct relevance to the current examination should not be coded, or should be indicated by a different code.

Since the ICD-9-CM codes are mainly used for billing purposes, the task itself is commercially relevant: false negatives (i.e. missed codes that should have been coded) will cause a loss of revenue to the health institute, while false positives (overcoding) is penalised by a sum three times higher than that earned with the superfluous code, and also entails the risk of prosecution to the health institute for fraud.

7.1 The corpus

For the experiments in automated ICD-9 coding of clinical free texts we used a publicly available dataset that contains radiology reports annotated with ICD-9-CM clinial codes.

This corpus was used for a shared task challenge on classifying clinical free text using NLP methods. The dataset consists of a training and a test part, a detailed description of the corpus texts and the annotation process can be found online at http://www.computationalmedicine.org/challenge/index.php and in [31]. The main characteristics of the dataset are summarised in the following table:

	Train	Test
Documents	978	976
Sentences	3225	3158
No. of codes	1218	1205

Table 7.1: Characteristics of the ICD-9-coding dataset.

7.2 Rule-based ICD-9-CM coders

Forty four teams submitted well-formatted results to the CMC'2007 challenge and, among the top performing systems, several exploited the benefits of expert rules that were constructed either by experts in medicine, or by computer scientists. This was probably due to the fact that reasonable well-formatted annotation guides are available online for ICD-9-CM coding and that expert systems can take advantage of such terms and synonyms that are present in an external resource (e.g. annotation guide or dictionary). Statistical systems on the other hand require labeled samples to incorporate medical terms into their learnt hypothesis and are thus prone to corpus eccentricities and usually discard infrequent transliterations or rarely used medical terms.

While the CMC challenge involved a considerable but limited number of codes (there were 45 distinct labels used in the challenge dataset), the feasibility of constructing expert systems for hundreds or thousands of codes is questionable and undoubtedly time consuming, especially if one wants to model all the possible inter-dependencies between labels.

7.2.1 Building an expert system from online resources

There are several sources from where the codes of the International Classification of Diseases can be downloaded in a structured form, including [90], [91] and [92]. Using one of these a rule-based system which performs ICD-9-CM coding by matching strings found in the dictionary to identify instances belonging to a certain code can be generated with minimal supervision. Table 7.2 shows how expert rules are generated from an ICD-9-CM coding guide. The system of Goldstein et al. [93] applies a similar approach and incorporates knowledge from [92].

These rule-based systems contain simple if-then rules to add codes when any one of the synonyms listed in the ICD-9-CM dictionary for the given code is found in the text, and removes a code when any one of the excluded cases listed in the guide is found. For example, code *591* is added if either *hydronephrosis*, *hydrocalycosis* or *hydroureteronephrosis* is found in the text and removed if *congenital hydronephrosis* or *hydroureter* is found. These expert systems – despite having some obvious deficiencies – can achieve a reasonable accuracy in labeling free text with the corresponding ICD-9-CM codes. These rule-based classifiers are data-independent in the sense that their construction does not require any labeled examples.

CODING GUIDE	GENERATED EXPERT RULES
label 518.0	**if document contains**
Pulmonary collapse	*pulmonary collapse* OR
Atelectasis	*atelectasis* OR
Collapse of lung	*collapse of lung* OR
Middle lobe syndrome	*middle lobe syndrome*
Excludes:	**AND document NOT contains**
atelectasis:	
congenital (partial) (770.5)	*congenital atelectasis* AND
primary (770.4)	*primary atelectasis* AND
tuberculous, current disease (011.8)	*tuberculous atelectasis*
	add label 518.0

Table 7.2: Generating expert rules from an ICD-9-CM coding guide.

7.3 Discovering inter-label dependencies

An important point which had to be dealt with, to get high performance coding systems from the rule-based systems described above, proved to be their lack of knowledge about inter-label dependencies needed to remove related symptoms when the code of a disease was added. Thus our goal here was to substitute the laborious process of manually discovering inter-label dependencies from labeled data with training machine learning models.

In order to discover relationships between a disease/illness and symptoms that arise from it, we applied statistical learning methods. For example, the presence of code *486* corresponding to *pneumonia* implied that the patient has certain symptoms like *786.2* and *780.6* (referring to *coughing* and *fever*). Coding systems that lack such knowledge regularly overcode documents with symptom labels. This kind of overcoding appears in the form of false positive symptom labels in the output of the coding system. This overcoding can be avoided by adding decision rules to the system to delete some symptom labels when evidence on certain diseases are found. These extra decision rules can be created manually. We found four rules good enough to worth adding to a manually constructed rule-based system (after manual inspection of the data). These were:

- delete code *786.2 (coughing)* when code *486 (pneumonia)* is present,
- delete code *780.6 (fever)* when code *486 (pneumonia)* is present,
- delete code *786.2 (coughing)* when code *493.90 (asthma)* is present and
- delete code *780.6 (fever)* when code *599.0 (urinary tract infection)* is present.

Deriving such rules based on observations of the data itself is actually quite time-consuming, so we decided to test whether or not such rules could be induced automatically. To perform the extraction of the above described disease-symptom relations, we first had to formulate the task as a classification problem for which we could utilise the labeled dataset, containing manual ICD-9 classifications. Since the annotation has been performed by physicians according to the official coding guidelines, the codes have been added to the documents with respect to the possible disease-symptom relations. A straightforward approach for discovering these relations is to train classifiers using the Vector Space representation of documents as features. This way, if a relation is statistically relevant (several examples are on hand that prove this relationship) the statistical model learns to avoid adding symptom labels when terms referring to a related disease are present (i. e. the model learns to *add label for coughing if the term 'cough' is found in the text, unless 'pneumonia' is present*). A major drawback of this approach is that it is prone to data sparseness as the classifier has to learn similar relations between each possible synonym pairs (including less frequent terms) referring to the same disease and symptom.

Our idea here was to try learning these relationships between classes rather than between medical terms. To do this, we first performed a pre-labeling by a simple ICD-9 coder derived from online coding guide sources. This approach, assuming that the coding system which performs pre-labeling has a full list of medical terms needed to identify each disease and symtom, would transform the problem of discovering inter-category dependencies with perfect precision. Of course the vocabularies of the coding system we applied were incomplete for many labels, as online guides list just the most common and normalised spellings. This fact meant that the task had a certain amount of noise associated with it.

This two-phase learning model for discovering inter-label dependencies actually performs a feature space transformation, which collapses all atomic features (single token n-grams) that refer to the same ICD category (different namings, spellings, abbreviations, etc. of the same concept) to a single, complex feature that symbolises the corresponding ICD category. This tranformation has a motivation similar to Latent Semantic Analysis, and is carried out by labeling the data with a statistical or rule-based ICD-9 coding system. The most important advantage of this approach compared to using atomic features is that it gave us the chance to detect dependencies at the label level instead of at the terminology level (as in the case when we used a uni-, bi- or trigram VSM representation).

In order to detect dependencies at the label level, we used the labels assigned by the initial rule-based system as features and trained a C4.5 decision tree classifier for each symptom label, treating the symptom false positive labels as the positive class and all other cases as negative examples. This way the decision tree learned to distinguish between false positive symptom labels and true positive ones. This statistical approach found five meaningful decision rules in the dataset, among which were all four rules that we enumerated above. The new rule was:

- delete code *788.30 (incontinence)* when code *593.70 (vesicoureteral reflux)* is present.

This fifth rule did not bring any improvement on the challenge test set (these two codes were never added to the same document). Because the four useful rules and the additional one that brought only a marginal improvement on the training dataset were found via our statistical approach – without inducing any detrimental disease-symptom relationships – we can say that this step of creating ICD-9-CM coding systems can be successfully automated. The modeling of inter-label dependencies brought about a 1.5% improvement in the performance of our rule-based system, raising the micro-averaged $F_{\beta=1}$ score from 84.07% to 85.57% on the training dataset and from 83.21% to 84.85% on the challenge test set.

7.4 Detailed comparison of performances via inter-annotator agreement rates

The results reported here are close to the performance that human expert annotators would achieve for the same task. The gold standard of the CMC challenge dataset is the majority annotation of three human annotators. The inter-annotator agreement statistics are shown in the following table:

	A1	A2	A3	GS	BasicRB	Hybrid
A1	–	73.97/75.79	65.61/67.28	83.67/84.62	75.11/75.56	78.02/79.19
A2	73.97/75.79	–	70.89/72.68	88.48/89.63	78.52/78.43	83.40/82.84
A3	65.61/67.28	70.89/72.68	–	82.01/82.64	75.48/74.29	80.11/78.97
GS	83.67/84.62	88.48/89.63	82.01/82.64	–	85.57/84.85	90.26/88.93
BasicRB	75.11/75.56	78.52/78.43	75.48/74.29	85.57/84.85	–	–
Hybrid	78.02/79.19	83.40/82.84	80.11/78.97	90.26/88.93	–	–

Table 7.3: Inter-annotator agreement rates between the annotators, the gold standard labeling ang two of our systems (the basic rule-based system and the best hybrid model).

We should mention here that the human annotators had no access to knowledge about the majority labeling, while models trained on the challenge dataset could model majority labeling directly. Thus, human annotator agreement with majority codes could have been higher if they had had the chance of examining the characteristics of majority labeling. On the other hand, the annotators influenced the target labels as these were created based on their own individual annotation. This fact explains why the annotators had a higher agreement rate with majority annotation than with the other human annotators. It would be interesting to see the agreement rate of a fourth human annotator and majority codes, given that the human annotator could then examine the characteristics of the majority codes but have no direct effect on their assignment. This statistic would provide a better insight into the theoretical upper bound for the system performance (the human performance) on this task.

The significantly lower agreement between individual human annotators shows that different health institutes probably have their own particular style of ICD-9-CM labeling. We also listed the agreement rates of annotators and the gold standard labeling with our basic rule-based system with label dependencies. This system can be regarded as a hypothetical human annotator in the sense that it models the ICD-9-CM coding guide an annotator should follow, not the gold standard labeling of the data itself. The fact that human annotators agree slightly better with this system than with each other also proves that they tend to follow specific standards that are not neccessarily confirmed by supported annotation guidelines. It is also interesting to see that majority labeling has a significantly higher agreement with this system than with individual annotators. This observation seems to justify that majority coding of independent annotators indeed estimates ICD-9-CM coding guidelines better than single expert annotators.

All the above findings apply when we restrict the agreement evaluation to the 45 labels that appear in the gold standard. Agreement between human annotators remains comparable to their agreement with the the coding guide (basic rule-based, BRB system). Each of the annotators has one preferred partner with whom their agreement is slightly better than with the BRB system, and each shows a lower agreement with the other human annotators. The gold standard labeling agrees better with BRB than any single annotation by almost 3%, which also indicates that majority annotation is capable of correcting mistakes and is better than any particular human annotation.

7.5 Results

Table 7.4 provides an overview of our results. All values are micro-averaged $F_{\beta=1}$. The *45-class statistical* row stands for a C4.5 classifier trained for single labels. The *CMC challenge best system* gave the results of the best system that was submitted to the CMC challenge. All our models used the same algorithm to detect negation and speculative assertions, and were trained using the whole training set (simple rule-based model needs no training) and evaluated on the training and the challenge test sets. The difference in performance between the 45-class statistical model and our best hybrid system (that is, using rule-based + MaxEnt models) proved to be statistically significant on both the training and test datasets, using McNemar's test with a $p < 0.05$ confidence level. On the other hand, the difference between our best hybrid model (constructed automatically) and our manually constructed ICD-9-CM coder (the CMC challenge best system) was not statistically significant on either set.

	train	test
45-class statistical	88.20	86.69
Simple rule-based	84.07	83.21
Rule-based with label-dependencies	85.57	84.85
Hybrid rule-based + C4.5	90.22	88.92
Hybrid rule-based + MaxEnt	90.26	88.93
CMC challenge best system	90.02	89.08

Table 7.4: Overall results obtained from using different classifier models.

7.6 Comparison and conclusions

The possibilities of automating the ICD-9-CM coding task have been studied extensively since the 1990s. Larkey and Croft [94] assigned labels to full discharge summaries having long textual parts. They trained three statistical classifiers and then combined their results to obtain a better classification. Lussier et al. [95] gave an overview of the problem in a feasibility study. Lima et al. [96] took advantage of the hierarchical structure of the ICD-9 code set, a property that is less useful when only a limited number of codes is used, as in our study.

Automating the assignment of ICD-9-CM codes for radiology records was the subject of a shared task challenge organised by the Computational Medicine Center (CMC) in Cincinatti, Ohio in the spring of 2007. A detailed description of the task, and the challenge itself, can be found in [31], and also online [97].

The most recent results are clearly related to the 2007 Challenge on Classifying Clinical Free Text, and some of the systems that have been published so far can be found in [98], [99], [100] and [93].

An analysis of classification errors revealed that our results with the hybrid approach are quite close to the upper limit of performance that can be attained using the CMC challenge dataset. Thus the rule-based models and its statistical extensions described above proved to be very efficient building blocks of a high performance clinical coding system, while they address some clear deficiencies of building expert systems manually that can provide similar accuracy. The vast majority of classification errors are caused by very rare cases (single specific usages not covered).

7.7 Summary of Thesis results

The main results of this chapter can be summarised as follows. Together with his collegues, the author participated in the 2007 CMC shared task challenge on automated ICD-9-CM coding of medical free texts using Natural Language Processing. The major steps of the development of the system as a whole that was submitted to the challenge, and the results achieved are a shared and indivisible contribution of the co-authors.

In particular, the author made a major contribution

- to the development of a basic and an entirely hand-crafted rule-based classifier,

- to the design, implementation and interpretation of the complex inter-annotator agreement analysis and

- to the design of the machine learning model for discovering inter-label dependencies from the labeled corpus.

The basic rule-based system provided the basis of further development and experiments on applying ML techniques to the problem.

The agreement analysis provided a baseline performance (basic rule-based system) and a theoretical upper bound for comparisons (entirely hand-crafted rule-based system) and background for evaluation and conclusions.

In order to discover inter-label dependencies from the labeled corpus, the author developed a two-step learning approach which implemented a feature space tranformation by collapsing atomic keyword features to complex ones representing single ICD-9 categories. This approach resulted a model that was less prone to data sparseness and more robust on previously unseen data, and it attained a performance equivalent to one achieved by human processing.

These contributions were helpful in designing a complete theoretical model of constructing hybrid rule-based/machine learning systems for ICD-9-CM coding that exploit both online knowledge sources and labeled data. This methodology was described in [43], while our main conclusions of the challenge and a description of the development of our system will be published together with the challenge organisers in a future study.

Chapter 8

Appendix

Corpus

A corpus is a dataset of documents (textual data). The dataset might contain annotation with the text (annotated corpus) or only raw text (unlabeled corpus). For example, an annotated Named Entity corpus contains documents from some source and annotation for Named Entities (names of companies, geographical locations, person names, etc.).

Classifier

A classifier is a model that enables the classification of objects to one of some predefined classes. In this thesis we distinguish between rule-based classifiers (where the classification rules are manually constructed by a human expert) and statistical classifiers (where the classification is performed by a statistical model trained on a set of pre-classified examples).

C4.5[1]

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of Information Entropy. The training data is a set $S = s_1, s_2, \ldots$ of already classified samples. Each sample $s_i = x_1, x_2, \ldots$ is a vector where $x_1, x_2, \ldots$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1, c_2, \ldots$ where $c_1, c_2, \ldots$ represent the class that each sample belongs to.

C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized Information Gain (difference in entropy) that results from choosing an attribute for splitting the

[1] Source: Wikipedia http://en.wikipedia.org/wiki/C4.5_algorithm

data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sublists.

This algorithm has a few base cases, the most common base case being when all the given samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class. It might also happen that none of the features give you any information gain, in which case C4.5 creates a decision node higher up the tree using the expected value of the class. It also might happen that you have never seen any instances of a class; again C4.5 creates a decision node higher up the tree using expected values.

AdaBoost[2]

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire. It is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favour of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. Otherwise, it is less susceptible to the overfitting problem than most learning algorithms.

AdaBoost calls a weak classifier repeatedly in a series of rounds $t = 1, \dots, T$. For each call a distribution of weights Dt is updated that indicates the importance of examples in the data set for the classification. On each round, the weights of each incorrectly classified example are increased (or alternatively, the weights of each correctly classified example are decreased), so that the new classifier focuses more on those examples.

Maximum Entropy Classifier

Maximum Entropy Models [32] seek to maximise the conditional probability of classes, given certain observations (features).

This is performed by weighting the features in such a way that it maximises the likelihood of the observed labeling (the training data) being generated by our exponential model. In other words, the maxent model is the unique probability distribution which has maximum entropy subject to the constraints. This model does not incorporate any prior assumption into the model (the probability distribution).
If the constraints are given in the form of feature expectations (calculated using the training data):

$$\textstyle\sum_x p(x) f_i(x) = \alpha_i,$$

searching for the probability distribution with maximum entropy yields the following optimisation problem:

$$max H(p(x)) = -\textstyle\sum_x p(x) \ln p(x),$$

[2] Source: Wikipedia http://en.wikipedia.org/wiki/Adaboost

subject to the constraints and to $\sum_x p(x) = 1$ (that is, the resulting model is a valid probability distribution).

Multi-Layer Perceptron[3]

This class of networks consists of multiple layers of computational units, interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function.

The universal approximation theorem for neural networks states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron with just one hidden layer. This result holds only for restricted classes of activation functions, e.g. for the sigmoidal functions.

Multi-layer networks use a variety of learning techniques, the most popular being back-propagation. Here, the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques, the error is then fed back through the network. Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small. In this case, one would say that the network has learned a certain target function. To properly adjust weights, one applies a general method for non-linear optimisation that is called gradient descent. For this, the derivative of the error function with respect to the network weights is calculated, and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason, back-propagation can only be applied on networks with differentiable activation functions.

Support Vector Machines[4]

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalised linear classifiers. A special property of SVMs is that they simultaneously minimise the empirical classification error and maximise the geometric margin; hence they are also known as maximum margin classifiers.

Viewing the input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyperplane in that space, one which maximises the *margin* between the two data sets. To calculate the margin, we construct two parallel hyperplanes, one on each side of the separating one, which are "pushed up against" the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighbouring datapoints of both classes. The hope is that,

[3] Source: Wikipedia http://en.wikipedia.org/wiki/Feedforward_neural_network

[4] Source: Wikipedia http://en.wikipedia.org/wiki/Support_vector_machine

the larger the margin or distance between these parallel hyperplanes, the better the generalization error of the classifier will be.

One can create non-linear classifiers by applying the kernel trick (originally proposed by Aizerman et al.[101]) to maximum-margin hyperplanes.[102] The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be non-linear in the original input space.

Sequence labeling

Classification problems usually assume that individual observations or input objects are disconnected and independent (independently and identically distributed (i.i.d.) examples).

NLP problems do not neccesarily satisfy this assumption and involve making many decisions which are mutually dependent on each other. More sophisticated learning and inference techniques are needed to handle such situations in general. Such NLP problems can viewed as sequence labeling where the goal is to predict the proper sequence of class labels (categories), given a sequence of input objects (a piece of text). A typical sequence labeling problem is when each token in a sentence is assigned a label (part of speech, named entity, syntactic label, etc.). Labels of tokens are highly dependent on the labels of other tokens in the sequence (the context), and particularly dependent on their neighbours (i.e. the surrounding words or local context).

Sequence labeling can be implemented using classification models and features describing the surrounding tokens using a moving window (sequence labeling as classification). Sometimes it is useful to add surrounding category values as features, but these values are unavailable unseen sequences and have to be substituted by the predictions of the classifier (online evaluation).

Probabilistic sequence models allow integrating uncertainty over multiple, interdependent classifications and collectively determine the most likely global assignment. Two standard probabilistic sequence labeling models are Hidden Markov Models (HMM) and Conditional Random Fields (CRF).

Online evaluation

In this thesis we use the term *online evaluation* to refer to sequence labeling with dynamic classification models where some of the features values are the categories of the surrounding words (preceding or subsequent words, depending on the direction of processing). These feature values are computed on the fly during the processing of unlabeled data, as they are dependent on the evaluation of preceding objects (by the same model). In particular we used the Named Entity codes of preceding tokens as such dynamic features. Clearly, these kind of features can be readily used in training statistical models where a set of labeled sequences (sentences) is available. On the other hand, information on the previous lexical items is unavailable when the model is tested on previously unseen examples, thus the values of these dynamic attributes have to be computed during the evaluation, based on predictions for previous instances.

Vector Space Model[5]

Vector space model (or term vector model) is an algebraic model for representing text documents (and any objects in general) as vectors of identifiers, like index terms. It is used in information filtering, information retrieval, indexing and relevancy rankings.

Each document is represented as a vector and each dimension corresponds to a separate term. If a term occurs at least once in the document, its value in the vector will be non-zero. Several different ways of computing these values, also known as (term) weights, have been developed. One of the best known schemes is tf-idf weighting (see the example below).

The definition of a term depends on the application. Typically terms are single words, keywords, or longer phrases. If the words are chosen to be the terms, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus).

TF-IDF[6]

The tf-idf weight (term frequency-inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure employed to evaluate how important a word is for a document in a collection or corpus. The importance increases proportionally with the number of times a word appears in the document, but is compensated for the frequency of the word in the corpus.

The term frequency in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. The usual formula is

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}},$$

where $n_{i,j}$ is the number of occurrences of the given term in document d_j, and the denominator is the number of occurrences of all terms in document d_j.

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|},$$

[5] Source: Wikipedia http://en.wikipedia.org/wiki/Vector_space_model

[6] Source: Wikipedia http://en.wikipedia.org/wiki/Tf-idf

with

$|D|$: the total number of documents in the corpus

$|\{d_j : t_i \in d_j\}|$: the number of documents where the term ti appears (that is $n_{i,j} \neq 0$).

Then

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i.$$

Gazetteer

In this thesis we use the term *gazetteer* to denote specific lists of Named Entities (e.g. a gazetteer of location names contains geographical entities like Country, City, State names, names of Mountains, Rivers and Seas). This is loosely related to the common use of the word gazetteer which originally means a list (dictionary or directory) of geographic names, along with some additional information like GPS coordinates and population.

Dictionary

In this thesis we use the term *dictionary* to denote specific lists of terms that behave in a similar way grammatically and are useful for Named Entity detection and categorisation. E.g. a dictionary of company types lists different organization form denominators (like *Corp.*, *Ltd.* and *Association*) in multiple languages. This term is used to differentiate between these kinds of lists and lists of full named entities (these are called gazetteers in this study).

Part of Speech

Part of Speech (POS, lexical category) is a linguistic category of lexical items, which is generally defined by the syntactic or morphological behaviour of the lexical item in question. A POS code thus defines the morphological and basic syntactic properties of a lexical item in the text.

Syntax information

Syntax in general denotes the set of rules of how lexical items can be arranged to form a valid, meaningful sentence in a language. Syntactic information thus describes the grammatic roles and structure of natural language sentences. Chunk codes describe a shallow syntactic structure, identifying just the major grammatical constituents of the sentence (noun phrases, verb phrases, etc.).

Evaluation metric

An evaluation metric is a measure of goodness which is used to evaluate, compare and analyse the performance of an algorithm.

Accuracy

Accuracy is defined as the number of properly classified instances over the number of instances that have been evaluated.

True positive

We consider a system output as a *true positive* prediction (corresponding to a target class or target set of classes) in the case when the instance belongs to the target class and the system correctly predicts the target label (assigns the instance to the proper class).

False positive

We consider a system output as a *false positive* prediction (corresponding to a target class or target set of classes) in the case when the instance does not belong to the target class and the system wrongly predicts the target label to be in the target class (assigns the instance to the target class).

False negative

We consider a system output as a *false negative* prediction (corresponding to a target class or target set of classes) in the case when the instance belongs to the target class and the system wrongly predicts the target label to be in a different class (assigns the instance to another class).

True negative

We consider a system output as a *true negative* prediction (corresponding to a target class or target set of classes) in the case when the instance does not belong to the target class and the system correctly predicts the target label to be different from that of the target class (assigns the instance to another class).

Precision

Precision measures how precisely a system predicts a target class (or set of classes), that is,

$$Prec = \frac{True\,positives}{True\,positives + False\,positives}.$$

Recall

Recall measures the ratio of instances of a class (or set of classes) that the system actually recognises as members of the class in question. That is,

$$Rec = \frac{True\,positives}{True\,positives + False\,negatives}.$$

F measure

F measure is defined as the harmonic mean of precision and recall and measures the performance of a system with respect to a target class (or set of classes). That is,

$$F = \frac{2*P*R}{P+R}.$$

We note here that if we consider every possible class for evaluation, then the F measure, Precision, Recall and Accuracy are all equivalent. Using the harmonic mean for the aggregation of precision and recall, the F measure favors balanced systems (having acceptable precision and recall scores as well).

Weighted F measure

In some applications it is useful to differentiate between the importance of precision and recall (the definition immediately above considers precision and recall to be equally important). For example, if our goal is to retrieve a few relevant documents, precision should be given a higher weight in the evaluation process (to prefer systems that do not provide false positive answers). On the other hand, if it is important to retrieve each relevant document from a collection recall should be given a higher weight. A generalised, weighted formula of the F measure for any non-negative real β value is:

$$F = \frac{(1+\beta^2)*P*R}{\beta^2*P+R}.$$

For example, $\beta = 2$ weights recall twice as much as precision, while $\beta = 0.5$ gives twice as big a weight to precision as it does to recall.

Precision-Recall Curve, PRC

The Precision-Recall Curve for a given class is the graphical plot of Precision vs. Recall values for various confidence thresholds. That is, a classifier assigns a posterior probability p for each instance representing how likely it is that the instance in question belongs to the class. A separate Precision and Recall value can be calculated for any probability threshold used for assigning instances to the class. PRC simply plots Precision and Recall values for $0 \leq p \leq 1$.

Break-Even-Point, BEP

BEP is the point on the precision-recall curve of $spec$ class where $P = R$. If an exact $P = R$ cannot be realised due to the equal ranking of many instances, we use the point closest to $P = R$ and set

$$BEP = \frac{P+R}{2}.$$

BEP is an interesting metric because it demonstrates how well we can trade off precision for recall.

Inter-Annotator agreement

We calculate the agreement rate or consistency of two annotations as the F measure of one of the labelings, using the other annotation as gold standard reference. We call this value the agreement rate of the two annotators. We note here that the inter-annotator agreement rate is symmetric because transposing the role of the two annotations only permutes Precision and Recall values, but the harmonic mean (that is, the F measure) is the same in both cases.

Dragon toolkit

The Dragon Toolkit[103] is an open source, general purpose Natural Language processing toolkit written in Java. It is intended for academic use and consists of many built-in low level (segmentation, tokenisation, lemmatisation, etc.) and high level (named entity recogniser, biological term extractor, etc.) text processing services. We used the Dragon Toolkit for pre-processing texts (i.e. token extraction and lemmatisation) here.

Link Parser

The Link Parser[76] is an open source syntactic parser for English. We used it to parse English texts to generate features for text classification.

WEKA

The WEKA package[104] is an open source collection of various machine learning algorithms and statistical data processing (filtering, transformation, etc.) methods, written in Java. We used the implementations in WEKA for the majority of the experiments described in the thesis.

OpenNLP MaxEnt

OpenNLP Maxent package is an open-source Maximum Entropy Modeling toolkit in Java. We utilised the OpenNLP MaxEnt package in experiments with MaxEnt classifiers, as it is computationally more efficient than the WEKA logistic regression implementation.

Mallet

MALLET[105] is an integrated collection of Java code that is useful for statistical natural language processing, document classification, clustering, information extraction, and other machine learning applications. We used the Mallet package in experiments with Conditional Random Fields for a sequence labeling approach to NER using the feature representation we developed. These experiments were not elaborated on in the Thesis.

NER evaluation metrics

Typical errors of a NER system can be classified to the following categories:

1. Classifying a non-entity token as a Named Entity.
2. Completely missing an entity, i.e. classifying it as non-entity.
3. Identifying the phrase boundaries of an NE, but assigning it to a wrong category.
4. Correctly classifying an entity but failing to determine its boundaries.
5. Incorrect boundaries and an incorrectly assigned class label.
6. Correctly classifying a part (or parts) of an entity and incorrectly classifying another part or parts.

Examples of each error category are given in the following table:

error type	example
1	– ***On***$_{ORG}$ *the 1st of May.* – *The* ***Internet***$_{ORG}$ *is a useful source of information.*
2	– ***Bush*** *is infamous for Iraq.* – *He is a quarterback of* ***Red Sox****.*
3	– *She is attending to the* ***Johns***$_{PER}$ ***Hopkins***$_{PER}$*.* – ***Barcelona***$_{LOC}$ *has to play a qualifier for CL next season.*
4	– ***Next***$_{ORG}$ ***Wednesday***$_{ORG}$ ***Manchester***$_{ORG}$ *plays the final against Chelsea in Moscow.* – ***Sheffield***$_{ORG}$ ***Wednesday*** *promoted to Premier League with a 3-0 win against QPR.*
5	– ***Johnnie***$_{PER}$ ***Walker*** *increased its yearly sales to over 120 million bottles per year with the highly successful Formula 1 advertisement campaign.* – ***FC Barcelona***$_{LOC}$ *player Edmílson signed for Villareal.*
6	– ***Manchester***$_{LOC}$ ***United***$_{ORG}$ *is the richest football club in the world, according to a recent survey.* – ***Henry***$_{PER}$ ***Ford***$_{ORG}$ *(July 30, 1863 - April 7, 1947) was the father of modern assembly lines used in mass production.*

Table 8.1: Examples of different NER error categories.

Different NE evaluation metrics handle/weight different error types and different NE types differently. The three most widely used NE evaluation standards are those introduced for the MUC conferences (Which are referred to as the MUC evaluation), the CoNLL conferences (CoNLL or exact match evaluation) and the ACE evaluations.

MUC evaluation

In the MUC evaluation finding the correct type and correct phrase boundaries is scored separately. Type is considered as a true positive classification, if the system assigns the same category to a phrase as the gold standard, as long as there is an overlap between the labeled and the gold standard phrase (i.e. they share at least one tagged token and with the similar label). The system receives credit for a correctly identified phrase boundary if it properly locates all tagged tokens in a phrase, regardless of whether it assigned the proper label to the phrase. True Positive, False Positive and False Negative classifications are computed for finding both the correct type (classification) and boundaries (recognition) separately and finally these are summed. The final system score is the equi-weighted F measure of the summed Precision and Recall scores.

CoNLL evaluation

CoNLL evaluation credits systems just for correct recognition of full entities (i.e. correct recognition of the whole phrase and correct classification of each tokens on the phrase). This means that a single classification error like in *North*$_{ORG}$ *West*$_{MISC}$ *Company*$_{ORG}$ can result in several system errors (1 False Negative, the missed *North West Company* organization and 3 False Positives (2 organization FP phrases: *North*$_{ORG}$ and *Company*$_{ORG}$ and 1 miscellaneous FP phrase: *West*$_{MISC}$ in this case.). System performance here is also the equi-weighted F measure of Precision and Recall scores.

The underlying hypothesis behind the CoNLL-style evaluation – which is quite strict compared to the MUC evaluation for example – is that real systems can only benefit from correctly recognised and classified full phrases. This means that there is no sense in giving partial credit to partial matches. In experiments this strict evaluation caused a drop in the state-of-the-art performance from 95% (MUC) to about 87-89% (CoNLL).

In this thesis we used the CoNLL-style evaluation in our experiments in Named Entity Recognition.

Automatic Content Extraction (ACE[106]) evaluation

ACE evaluations targeted not just the detection and classification of proper name references to entities, but also nominal and pronominal references. This means that the scope of ACE evaluation campaigns is wider than previous NER evaluations (CoNLL and MUC). Entity Detection and Recognition scoring also involves coreference resolution; that is, each entity mention has to be assigned to the corresponding unique entity. By incorporating coreference resolution and detecting nominal and pronominal references, ACE campaigns go one step beyond the scope of previous evaluation styles (and thus the scope of thesis, since we restricted ourselves to CoNLL style task definition and evaluation). A detailed description of the ACE evaluation can be found online at: http://www.nist.gov/speech/tests/ace/2007/doc/.

Chapter 9

Summary

Summary in English

Introduction

In this thesis we presented from a feature representation point of view, several practical text mining applications developed together with colleagues. The applications themselves cover a wide range of tasks from entity recognition (word sequence labelling) to multi-label document classification and also cover different domains (from business news texts to medical records / biological scientific articles). Our aim was to demonstrate that task-specific feature engineering is beneficial to the overall performance and for specific text mining tasks one can construct systems that are useful in practice and even compete with humans in processing textual data.

The summary below, like the thesis itself, consists of two main parts, each addressing an important topic in Text Mining. The first part summarises our findings in Named Entity Recognition problems and in the second part we describe work done in Text Classification.

Named Entity Recognition

The identification and classification of rigid designators like proper nouns, acronyms of names, time expressions, measures, IDs, e-mail addresses and phone numbers. in plain text is of key importance in numerous natural language processing applications. The special characteristic of these rigid designators (as opposed to common words) is that they have no meaning in the traditional sense but they refer to one or more entities of the world in a unique way (references). In the literature these text elements are called Named Entities (NEs).

In Named Entity Recognition (NER) problems, one tries to recognise (single or subsequent) tokens in text that together constitute a rigid designator phrase, and to determine the category type to which these phrases belong. Categorisation is always task specific, as different kinds of entities are important in different domains. Sometimes entity recognition itself can be a standalone application, as in the case of anonymisation issues, where no further processing is required when all the name phrases have been located in the text.

Results

Together with his collegues the author designed and developed a language and domain independent Named Entity Recognition framework that was successfully applied to many similar tasks like the recognition of *person, organization, location* names and other proper nouns in Hungarian short news articles and in English news articles. These results are presented in [33] and [34]. The same system was successfully adapted to a quite different domain and performed well in the recognition of *patient and doctor names, the age of the patient, phone numbers, IDs, locations, hospital names and dates* in medical discharge summaries in English. This system was entered in an open challenge on medical record de-identification and it achieved the second best score on a standard evaluation dataset. All the results are presented in [39]. In these studies the author was mainly responsible for the design and implementation of a feature representation that aided the learning models used and thus contributed to a good performance.

Later on the author investigated corpus frequency-based heuristics, which were capable of fine tuning NER systems by eliminating typical errors of NER systems. These heuristics were then modified to provide a heuristic solution to Named Entity lemmatisation, a problem that arises both in English (plural and possessive markers) and in Hungarian (agglutinative characteristic of the language). Since morphological analyser systems usually rely on a list of valid lemmas for a given language they are usually ill-suited for NE lemmatisation – as an exhaustive list of normalised NEs is impossible to gather in practice. The author and his collegues showed that corpus statistics can be utilised to handle NE lemmatisation and achieve a good accuracy. These results are presented in [36], [37]. In these articles the author was mainly responsible for the design of web-based heuristics for NE lemmatisation and the design and implementation of a heuristic method based on a Wikipedia search for separating consecutive NEs.

Text Classification

The human processing of textual data (system logs, medical reports, newswire articles, customer feedback records, etc.) is a laborious and costly process, and is becoming unfeasible with the increasing amount of information stored in documents. There is a growing need for solutions that automate or facilitate the information processing workflow that is currently being performed by humans. Thus today the automatic processing of free texts (either assertions or longer documents) based on their content and converting textual data to practical knowledge is an important subtask of Information Extraction.

Many text processing tasks can be formulated as a classification problem and solved effectively with Machine Learning methods that are capable of uncovering the hidden structure in free text, assuming that labelled examples are on hand to train the automatic systems on. These solutions go one step beyond simple information retrieval (that is, providing the user with the appropriate documents using keyword lookup and relevance ranking) as they require the (deep or shallow) understanding of the text itself. The systems have to handle synonymy, transliterations and language phenomena like negation, sentiment, subjectivity and temporality.

A major application domain of practical language technology solutions is in the fields of Biology and Medicine. Experts in these fields usually have to work with large collections of documents in everyday work in order to carry out efficient research (reading scientific papers, patents, or reports on earlier experiments in a topic) and asses data in decision-making processes (reports on the examination of former patients with similar symptoms or diseases, etc.).

Results

Together with his collegues the author designed and developed a system that classified medical records according to the patient's smoking status. This system was entered in an open challenge on smoking status classification in medical records and achieved competitive performance on a standard evaluation dataset. These results are presented in [40]. In this study the author was mainly responsible for the design and implementation of a feature representation that aided the learning models used and thus contributed to a good performance.

Later on together with his collegue the author designed and developed a system for the clinical coding of medical reports. This system was entered in an open challenge on the ICD-9-CM coding of radiology reports and achieved the best performance on a standard evaluation dataset, among all the systems entered in the challenge. Based on the results and lessons learned in the challenge, the author and his collegues designed a complex framework for combining expert knowledge with machine learning models that exploits labeled examples. This is the so-called hybrid (expert-rule based and statistical) approach to ICD-9-CM coding and was presented in [43]. In this study the author was mainly responsible for the design and implementation of a feature representation for discovering inter-label dependencies that contributed to a good performance. The author was also reponsible for the design and implementation of the rule-based system that served as the basis of further research and development, and for the realisation of a complex agreement analysis procedure between human experts and computer systems which helped provide a better insight into the value of the results and allowed us to make a meaningful comparison between the various approaches.

Finally the above problems motivated the author to focus his research interest on negation and speculation detection as these phenomena play a key role in the language of biomedicine (both scientific and clinical language). The importance of the accurate recognition of negative and uncertain findings is demonstrated by their huge impact on the performance of text mining applications in the biomedical domain. The author designed and implemented a complex feature (or keyword) selection method that combines statistical methods and expert supervision in order to extract meaningful hedge cues from examples having partly or entirely automatically generated labels with minimal or no supervision. The results for this are presented in [41]. Here the author was responsible for all the work described in the study. Besides the detection of meaningful negation and uncertainty cues, an important issue is to locate their scope in the sentence (because it is not always the whole sentence that is negated or uncertain and thus it might contain useful positive facts too). To overcome the second difficulty the author participated in the design and supervised the construction of a large scale annotated corpus for negations, speculations and their scope. This work resulted in an open source corpus that will, hopefully, facilitate research on scope detection and negation/speculation in general in the future. The details of this corpus project are given in [42].

Conclusions of the Thesis

All specific HLT problems discussed in the thesis fall within the field of Text Mining, with most of them sharing a common domain, i.e. that of biomedical language processing. What our solutions have in common is that

- we paid close attention to the feature engineering issues involved
- the vast majority of our features were discrete in nature, or had continuous values but a straightforward and meaningful discretisation was possible
- we used well-known classifiers like C4.5 decision trees and Maximum Entropy classifiers.

Our good results demonstrate that proper feature engineering can provide a representation where even simple learning models achieve competitive results. We partly attribute this to the characteristics of the feature space representation (i.e. we chose to design a compact representation instead of feature spaces of very high dimensionality) and the relative strength of C4.5 and MaxEnt in handling discrete (or binary) features. As these learning algorithms are especially suitable for binary/discrete valued features, the good performace scores we obtained is not really surprising. On the other hand, our models have some desirable characteristics in training and low processing time. Quick training and testing comes from the employment of learning methods with a moderate training time complexity and feature spaces that have a low dimension after incorporating as much information on the target variable as possible. From this we conclude that our solutions are potential candidates for use in the kind of situations where rapid training and testing time is a must, even at the cost of a longer development time spent on feature engineering. We always avoided the exhaustive use of simple token n-gram features, i.e. the Vector Space Model representation that often leads to very high dimensional feature spaces without the grouping and selection of n-grams.

Summary in Hungarian

Bevezető

A disszertációban számos gyakorlati szövegbányászati problémát, illetve azokra a kollégákkal közösen kifejlesztett rendszereket mutatunk be, a jellemzőtér-reprezentáció, illetve a jellemzőkinyerés oldaláról. A tárgyalt problémák a névelem-felismerés (szósorozatok címkézése) és a dokumentumosztályozás feladatkörébe tartoznak, illetve az alkalmazási területük is változatos: az üzleti hírektől a kórházi jelentésekig terjed. A disszertáció célja, hogy megmutassuk: a feladatspecifikus jellemzőkinyerési fejlesztések segítségével jó pontosságot érhetünk el, illetve az eredményül előálló rendszerek ezáltal a gyakorlatban is sikeresen alkalmazhatók (egyes esetekben még az emberi feldolgozás pontosságát is megközelítik az elemzett szövegek legnagyobb részén).

Az összefoglaló szerkezeti felépítése követi a disszertációét, azaz két főbb részre oszlik, melyek a Szövegbányászat egy-egy intenzíven kutatott részterületét képezik. Az első rész összefoglalja a névelem-felismeréshez kapcsolódó eredményeket, míg a második részben a dokumentumosztályozás területén végzett kutatásunk eredményeit ismerteti.

Névelem-felismerés

A szövegben található névelem-kifejezések (tulajdonnevek, nevek akronimjai, mennyiséget, időt jelölő kifejezések, azonosítók, e-mail címek, közigazgatási címek, telefonszámok, stb.) azonosítása és osztályozása a Szövegbányászat egyik legalapvetőbb feladata. Ez a kifejezések úgynevezett merev jelölők, melyeknek a köznyelvi szavakkal ellentétben nincs jelentésük, hanem a világ valamely entitására vagy egy csoportra egyedi módon hivatkoznak (egyfajta referenciák). Ezeket a merev jelölőket a szakirodalomban *névelem*eknek is nevezik.

A névelem-felismerés (Named Entity Recognition, NER) célja a szövegben olyan tokeneknek (vagy tokenek egymást követő sorozatainak) a megtalálása, melyek egy merev jelölő frázist alkotnak, majd a megtalált frázisok pontos kategorizálása. A kategorizáció mindig az adott alkalmazásra jellemző, hiszen más típusú entitások lényegesek az egyes feladatoknál. A gyakorlatban a szövegben megtalált névelemek további feldolgozás alapját képezik, azaz felismerésük egy köztes lépés a feldolgozási folyamatban, azonban néha maga a NER is lehet önálló végalkalmazás. Ilyen példa az anonimizálás, ahol a névelemek felismerése után csak azok eltávolítása vagy lecserélése történik, hogy a személyes adatoktól megtisztítsuk a dokumentumot.

Eredmények

A szerző és társai megtervezték és kifejlesztettek egy olyan, nyelv- és doménfüggetlen névelem-felismerő keretrendszert, amelyet több hasonló feladat megoldására eredményesen alkalmaztak. Ilyen feladat pl. a *személy, szervezet és helynevek* illetve egyéb tulajdonnevek felismerése és osztályozása magyar nyelvű üzleti rövidhírek szövegeiben, illetve angol nyelvű újsághírekben. Ezeket az eredményeket a [33] és [34] publikációk ismertetik. Ugyanezt a rendszert eredményesen alkalmazták egy lényegesen eltérő területen is, ahol az szintén kiemelkedő pontosságot ért el. Angol nyelvű kórházi dokumentumokban *betegek, orvosok neveit, a beteg életkorát, telefonszámokat, azonosítókat, helyneveket, kórházneveket és dátumokat azonosítottak*. Ez az orvosi dokumentumokon működő rendszer a második legjobb eredményt érte el egy anonimizáló rendszerek kiértékelésére szolgáló adatbázison. Ezeket az eredményeket a [39] publikáció ismerteti. Az említett három alkalmazásban a szerző elsődlegesen a jellemzőkinyerési fejlesztések megtervezésében és elvégzésében vállalt döntő szerepet.

Ezt követően a szerző korpuszgyakoriságon alapuló heurisztikák kifejlesztésén dolgozott, amelyek a névelem-felismerő rendszerek bizonyos tipikus hibáinak kijavítására voltak alkalmasak. Később ezeket a heurisztikus eljárásokat oly módon tervezte újra, hogy azok a névelemek normalizálására (szótövezés és inflexiók eltávolítása) általában is alkalmasakká váltak. A névelemek normalizálása mind angol, mind magyar nyelven kihívást jelentő feladat, hiszen a morfológiai elemzők a legtöbbször nem igazán működnek megbízhatóan, ha névelemek elemzésére használjuk őket. A szerző és társai kísérletekkel igazolta, hogy az általuk kidolgozott heurisztikus NE-normalizáló eljárás igen jó eredményt ad. Ezeket az eredményeket a [36] és [37] publikációk ismertetik.

Dokumentumosztályozás

A szöveges adatok (rendszer logok, orvosi jelentések, újságcikkek, fogyasztói visszajelzések, stb.) emberi feldolgozása munkaigényes és költséges feladat, ami a szöveges információ növekedtével egyre nehezebben oldható meg. Egyre növekszik az igény olyan megoldások iránt, amelyek automatizálják vagy felgyorsíthatják a most még sokszor emberek által végzett adatelemző, információkereső tevékenységet. Emiatt a természetes nyelvi szövegek automatikus kategorizálása/osztályozása napjainkra a Szövegbányászat egyik legfontosabb feladatává vált.

Sok szövegfeldolgozási feladat felírható a gépi tanulás területén közismert ún. osztályozási feladatként, ami lehetővé teszi azok gépi tanulási algoritmusok segítségével történő, eredményes megoldását. Ezek a megoldások képesek a folyó szövegben megtalálható rejtett szabályszerűségek, struktúra felfedezésére, amennyiben rendelkezésünkre állnak címkézett dokumentumok, melyek segítségével a rendszerek taníthatók. A dokumentumosztályozási feladatok esetén a rendszertől elvárt kimenet minden esetben használható tárgyi tudás (tényszerű információ), nem pedig egy döntés, hogy a dokumentum tartalmaz-e számunkra érdekes információt vagy sem. Emiatt ezek a megoldások általában túlmutatnak az egyszerű kulcsszavas információkeresési technikákon (kulcsszavas keresés és a találatok rangsorolása), hiszen a feladatok szükségessé teszik a szövegek bizonyos szintű *megértését*. A dokumentumosztályozó rendszereknek általában kezelniük kell a különböző írott alakok, a szinonímia, vagy pl. a tagadás, érzelmi töltet, bizonytalanság, illetve az időbeliség okozta nehézségeket.

A szövegbányászati megoldások legnagyobb alkalmazási területei közé tartozik a Biológia és a Gyógyászat. Az ezeken a területen dolgozó szakemberek, kutatók általában nagy mennyiségű szöveges dokumentummal dolgoznak a mindennapi munkájuk során a kutatásban (tudományos publikációkat, szabadalmakat, vagy a témához kapcsolódó korábbi kísérletek beszámolóit olvassák) vagy a döntéshozásban (pl. korábbi, hasonló tünetekkel, vagy diagnózissal kezelt betegek kórtörténetét elemzik).

Eredmények

A szerző és társai kifejlesztettek egy olyan dokumentumosztályozó modellt, amely a betegek kórházi zárójelentése alapján képes besorolni a beteget dohányzási szokásának megfelelő kategóriába. A kifejlesztett rendszer egy szövegbányászati kiértékelési versenyen jó eredményt ért el. A rendszer kidolgozása során a szerző hozzájárulása a jellemzőkinyerési munkák megtervezésében és kivitelezésében volt meghatározó. Ezeket az eredményeket a [40] publikáció ismerteti.

Később a szerző és társa egy kórházi leletek betegségkódokkal (Betegségek Nemzetközi Osztályozása, BNO-kódok) való automatikus címkézésére alkalmas rendszert fejlesztett ki. Ez a rendszer egy automatikus klinikai kódoló rendszerek kiértékelésére szervezett versenyen a legjobb pontosságot érte el. A verseny tapasztalatai alapján a szerző és társa egy szakértői és statisztikai rendszerek kombinációján alapuló modellt fejlesztett ki, mely képes a rendelkezésre álló szabályalapú rendszereket címkézett példák felhasználásával tovább pontosítani, fejleszteni. Ez az ún. hibrid megközelítés a [43] publikációban került bemutatásra. Az ide kapcsolódó fejlesztésekből a szerző hozzájárulása volt meghatározó a címkeközi összefüggések felderítésére alkalmas modellhez felhasznált jellemzőkinyerési munkák megtervezésében és megvalósításában, a kezdeti modellként a továbbfejlesztésekhez felhasznált szabályalapú rendszer kidolgozásában és megvalósításában. A szerző végezte el a modellek alapos összevetését lehetővé tevő komplex annotátor-egyetértési elemzést, illetve az annotátorok illetve egyes automatikus rendszerek összevetését (melyhez a felhasznált, tisztán szakértői szabályokon alapuló rendszer kifejlesztését is elvégezte).

Végül az alábbi feladatok a szerző érdeklődését két, a szövegbányászatban nagyon fontos nyelvi jelenség, a tagadás és a bizonytalanság (spekulációk) automatikus felismerésére terelték. E két nyelvi jelenség pontos felismerése és megfelelő kezelése döntő hatással van a biológiai, illetve klinikai szövegbányászati megoldások eredményességére, alkalmazhatóságára. A szerző megtervezett és tesztelt egy komplex jellemzőkiválasztási módszert, mely a statisztikai modellek és emberi tudás kombinációjával minimális ráfordítás mellett lehetővé tette az értelmes spekulatív kulcsszavak felismerését félig vagy teljesen automatikus módon gyártott tanítóadatbázisok használata mellett is (nem volt szükség emberi címkézésre). Ezeket az eredményeket a [41] publikáció ismerteti. Ez a publikáció teljes egészében a szerző saját eredményeit ismerteti. A megfelelő tagadó vagy spekulatív kulcsszavak megtalálása mellett nagyon fontos a kulcsszavak nyelvi hatókörének megállapítása is (tehát nem minden esetben a teljes mondat jelentése spekulatív, ha abban egy kulcsszó előfordul, hanem sokszor csak egy-egy mondatrész, tagmondat jelentése módosul). E második probléma megoldására a szerző részt vett egy, a negált és bizonytalan elemek és azok hatókörének bejelölését célzó korpuszannotációs projekt megtervezésében és felügyeletében. A projekt eredményeként elkészült az első, kutatási célokra szabadon hozzáférhető korpusz, amely remélhetőleg elősegíti majd a témához kapcsolódó további kutatási eredmények létrejöttét. Ezeket az eredményeket a [42] publikáció ismerteti.

Konklúzió

Minden a disszertációban bemutatott feladat a Szövegbányászat témaköréhez tartozik (legtöbbjük a biológiai vagy klinikai területről). Az általunk adott megoldások közös jellemzője, hogy

- jelentős energiát fektettünk a feladathoz jól használható jellemzők megtervezésébe és fejlesztésébe
- a legtöbb általunk használt jellemző vagy eleve diszkrét értékekkel leírható volt, vagy természetesen módon adódott egy diszkretizált felírása
- jól ismert és széles körben használt osztályozókat használtunk (C4.5 döntési fa, Maximum Entrópia osztályozó)

Az elért eredmények azt mutatják, hogy a megfelelő jellemzők használatával akár egyszerűbb algoritmusok is jó eredményt adhatnak. Ezt részben az általunk kifejlesztett reprezentáció kedvező tulajdonságainak tulajdonítjuk (igyekeztünk a lehető legkisebb dimenziójú jellemzőtérbe sűríteni az összegyűjtött információt), részben pedig annak, hogy a C4.5 illetve a MaxEnt osztályozók kimondottan alkalmasak diszkrét jellemzők tanulására.

Az általunk kifejlesztett modelleknek egy további előnyös tulajdonsága is van, a tanítási és tesztelési idő terén. Viszonylag alacsony időigényű tanulóalgoritmusokat használtunk, a lehető legkisebb dimenziójú jellemzőtér használatával, így modelljeink gyorsan taníthatók és feldolgozási sebességük is igen gyors. Ezáltal a kifejlesztett rendszerek jól alkalmazhatók olyan szituációkban, ahol a gyors tanítás és tesztelés elengedhetetlen, akár a hosszabb fejlesztési idő árán is (munkánk során kerültük az egyszerű szóalapú vektorteres reprezentáció használatát, legalábbis a megfelelő csoportosítás, egyszerűsítés és szelekció nélkül, hiszen ez kezelhetetlen méretű jellemzőteret és nagyobb időigényt eredményezett volna).

Bibliography

[1] Kripke S: *Naming and Necessity*. Harvard University Press 1972.

[2] Jurafsky D, Martin JH: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition 2008, [http://www.amazon.de/exec/obidos/redirect?tag=citeulike01-21\&amp;path=ASIN/013122798X].

[3] Babych B, Hartley A: **Improving Machine Translation Quality with Automatic Named Entity Recognition**. In *Proceedings of the 7th International EAMT workshop at EACL-2003*, Budapest, Hungary: Association for Computational Linguistics 2003:18–25.

[4] Nicolae C, Nicolae G: **BESTCUT: A Graph Algorithm for Coreference Resolution**. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia: Association for Computational Linguistics 2006:275–283, [http://www.aclweb.org/anthology/W/W06/W06-1633].

[5] Cucerzan S: **Large-Scale Named Entity Disambiguation Based on Wikipedia Data**. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* 2007:708–716.

[6] Chinchor NA: **Overview of MUC-7/MET-2.** In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* 1998.

[7] Tjong Kim Sang EF, De Meulder F: **Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:142–147.

[8] Tjong Kim Sang EF: **Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition**. In *Proceedings of CoNLL-2002*, Taipei, Taiwan 2002:155–158.

[9] Tou Ng H, Kwong OOY (Eds): *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, Sydney, Australia: Association for Computational Linguistics 2006.

[10] Sekine S, Isahara H: **IREX: IR and IE evaluation project in Japanese** 2000, [citeseer.ist.psu.edu/sekine00irex.html].

[11] Uzuner O, Luo Y, Szolovits P: **Evaluating the State-of-the-Art in Automatic De-identification**. *J Am Med Inform Assoc* 2007, **14**(5):550–563, [http://www.jamia.org/cgi/content/abstract/14/5/550].

[12] Corbett P, Batchelor C, Teufel S: **Annotation of Chemical Named Entities**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007[http://www.aclweb.org/anthology/W/W07/W07-1008].

[13] Kim J, Ohta T, Tsuruoka Y, Tateisi Y, Collier N: **Introduction to the bio-entity recognition task at JNLPBA**. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), Geneva, Switzerland*. Edited by Collier N, Ruch P, Nazarenko A 2004:70–75.

[14] Grishman R, Sundheim B: **Message Understanding Conference-6: a brief history**. In *Proceedings of the 16th conference on Computational linguistics*, Morristown, NJ, USA: Association for Computational Linguistics 1996:466–471.

[15] Cucerzan S, Yarowsky D: **Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence**. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, MD, USA: Association for Computational Linguistics 1999:90–99.

[16] Kozareva Z: **Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists**. In *Proceedings of the Student Research Workshop at 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy: Association for Computational Linguistics 2006:15–21.

[17] Lee HS, Park SJ, Jang H, Lim J, Park SH: **Domain Independent Named Entity Recognition from Biological Literature**. In *Proceedings of The 15th International Conference on Genome Informatics*, Yokohama, Japan 2004.

[18] Szarvas Gy, Farkas R, Felföldi L, Kocsor A, Csirik J: **A highly accurate Named Entity corpus for Hungarian**. In *Proceedings of Language Resources and Evaluation Conference* 2006.

[19] Csendes D, Csirik J, Gyimóthy T, Kocsor A: **The Szeged Treebank**. In *TSD* 2005:123–131.

[20] Quinlan JR: *C4.5: Programs for Machine Learning*. Morgan Kaufmann 1993.

[21] Schapire R: **The boosting approach to machine learning: An overview**. In *Proceedings of MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA, USA 2001.

[22] Sebastiani F: **Machine learning in automated text categorization**. *ACM Comput. Surv.* 2002, **34**:1–47, [http://portal.acm.org/citation.cfm?id=505282.505283].

[23] Shanahan JG, Qu Y, Wiebe J: *Computing Attitude and Affect in Text: Theory and Applications (The Information Retrieval Series)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. 2005.

[24] Ananiadou S, Mcnaught J: *Text Mining for Biology And Biomedicine*. Norwood, MA, USA: Artech House, Inc. 2005.

[25] Zeng Q, Goryachev S, Weiss S, Sordo M, Murphy S, Lazarus R: **Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system**. *BMC Medical Informatics and Decision Making* 2006, **6**:30, [http://www.biomedcentral.com/1472-6947/6/30].

[26] Uzuner O, Goldstein I, Luo Y, Kohane I: **Identifying Patient Smoking Status from Medical Discharge Records**. *J Am Med Inform Assoc* 2008, **15**:14–24, [http://www.jamia.org/cgi/content/abstract/15/1/14].

[27] Light M, Qiu XY, Srinivasan P: **The Language of Bioscience: Facts, Speculations, and Statements In Between**. In *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*. Edited by Hirschman L, Pustejovsky J, Boston, Massachusetts, USA: Association for Computational Linguistics 2004:17–24.

[28] Hyland K: **Hedging in Academic Writing and EAP Textbooks**. *English for Specific Purposes* 1994, **13**(3):239–256.

[29] Medlock B, Briscoe T: **Weakly Supervised Learning for Hedge Classification in Scientific Literature**. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic: Association for Computational Linguistics 2007:992–999, [http://www.aclweb.org/anthology/P/P07/P07-1125].

[30] Lang D: **Consultant Report - Natural Language Processing in the Health Care Industry**. *PhD thesis*, Cincinnati Children's Hospital Medical Center 2007.

[31] Pestian JP, Brew C, Matykiewicz P, Hovermale D, Johnson N, Cohen KB, Duch W: **A shared task involving multi-label classification of clinical free text**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:97–104, [http://www.aclweb.org/anthology/W/W07/W07-1013].

[32] Berger AL, Pietra SD, Pietra VJD: **A Maximum Entropy Approach to Natural Language Processing**. *Computational Linguistics* 1996, **22**:39–71, [citeseer.ist.psu.edu/berger96maximum.html].

[33] Farkas R, Szarvas Gy, Kocsor A: **Named entity recognition for Hungarian using various machine learning algorithms**. *Acta Cybern.* 2006, **17**(3):633–646.

[34] Szarvas Gy, Farkas R, Kocsor A: **A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms**. In *Discovery Science* 2006:267–278.

[35] Markert K, Nissim M: **SemEval-2007 Task 08: Metonymy Resolution at SemEval-2007**. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic: Association for Computational Linguistics 2007:36–41, [http://www.aclweb.org/anthology/W/W07/W07-2007].

[36] Farkas R, Szarvas Gy, Ormándi R: **Improving a State-of-the-Art Named Entity Recognition System Using the World Wide Web**. In *Industrial Conference on Data Mining* 2007:163–172.

[37] Farkas R, Vincze V, Nagy I, Ormándi R, Szarvas Gy, Almási A: **Web based lemmatisation of Named Entities**. In *Accepted for 11th International Conference on Text, Speech and Dialogue* 2008.

[38] Farkas R, Simon E, Szarvas Gy, Varga D: **GYDER: Maxent Metonymy Resolution**. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic: Association for Computational Linguistics 2007:161–164, [http://www.aclweb.org/anthology/W/W07/W07-2033].

[39] Szarvas Gy, Farkas R, Busa-Fekete R: **State-of-the-art anonymisation of medical records using an iterative machine learning framework**. *J Am Med Inform Assoc* 2007, **14**(5):574–580, [http://www.jamia.org/cgi/content/abstract/M2441v1].

[40] Szarvas Gy, Iván S, Bánhalmi A, Csirik J: **Automatic Extraction of Semantic Content from Medical Discharge Records**. *WSEAS Transaction on Systems and Control* 2006, **1**(2):312–317.

[41] Szarvas Gy: **Hedge classification in biomedical texts with a weakly supervised selection of keywords**. In *Accepted for the 45th Annual Meeting of the Association of Computational Linguistics*, Columbus, Ohio, United States of America: Association for Computational Linguistics 2008.

[42] Szarvas Gy, Vincze V, Farkas R, Csirik J: **The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts**. In *Accepted for Biological, translational, and clinical language processing (BioNLP Workshop of ACL)*, Columbus, Ohio, United States of America: Association for Computational Linguistics 2008.

[43] Farkas R, Szarvas Gy: **Automatic construction of rule-based ICD-9-CM coding systems**. *BMC Bioinformatics* 2008, **9**(3), [http://www.biomedcentral.com/1471-2105/9/S3/S10].

[44] Kuba A, Hócza A, Csirik J: **POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods**. In *TSD* 2004:113–120.

[45] Kozareva Z, Silva JF, Lopes GP: **Cluster Analysis and Classification of Named Entities**. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'2004*. Edited by Lino MT, Xavier MF, Ferreira F, Costa R, , ELRA - European Language Resources Association 2004:321–324. [URL=http://www.dlsi.ua.es/ zkozareva/papers/lrec2004ClustAna.pdf].

[46] Gábor K, Héja E, Ágnes Mészáros, Sass B: **Nyílt tokenosztályok reprezentációjának technológiája.** Tech. Rep. IKTA-00037/2002, Hungarian Academy of Sciences, Research Institute for Linguistics, Budapest, Hungary 2002.

[47] Prószéky G: **Syntax As Meta-morphology**. In *Proceedings of 16th International Conference on Computational Linguistics, COLING-96*, Association for Computational Linguistics 1996:1123–1126.

[48] Varga D, Simon E: **Hungarian named entity recognition with a maximum entropy approach**. *Acta Cybernetica* 2007, **18**(2):293–301.

[49] Rose T, Stevenson M, Whitehead M: **The reuters corpus volume 1-from yesterday's news to tomorrow's language resources**. *Proceedings of the Third International Conference on Language Resources and Evaluation* 2002, :29–31, [http://about.reuters.com/researchandstandards/corpus/LREC_camera_ready.pdf].

[50] Srihari RK, Niu C, Li W, Ding J: **A Case Restoration Approach to Named Entity Tagging in Degraded Documents**. In *Proceedings of 7th International Conference on Document Analysis and Recognition (ICDAR 2003)* 2003:720–724.

[51] Rau LF: **Extracting Company Names from Text**. In *Proceedings of the Seventh Conference on Artificial Intelligence Applications*, Miami Beach, FL 1991:189–194.

[52] Etzioni O, Cafarella M, Downey D, Popescu AM, Shaked T, Soderland S, Weld DS, Yates A: **Unsupervised named-entity extrac-**

tion from the web: an experimental study. *Artificial Intelligence* 2005, **165**:91–134, [http://www.sciencedirect.com/science/article/B6TYF-4FY3P4K-1/2/de4869f2336ce10c4dbeb0671d30d96a].

[53] Evans R: **A Framework For Named Entity Recognition in the Open Domain**. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2003)*, Borovetz, Bulgaria 2003:137 – 144, [http://clg.wlv.ac.uk/papers/evans-RANLP-03.pdf].

[54] Cimiano P, Völker J: **Towards large-scale, open-domain and ontology-based named entity classification**. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*. Edited by Angelova G, Bontcheva K, Mitkov R, Nicolov N, Borovets, Bulgaria: INCOMA Ltd. 2005:166–172, [\url{http://www.aifb.uni-karlsruhe.de/WBS/pci/Publications/ranlp05.pdf}].

[55] Nadeau D: **Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision**. *PhD thesis*, University of Ottawa 2007, [http://cogprints.org/5859/].

[56] Sekine S, Nobata C: **Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy**. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'2004*. Edited by Lino MT, Xavier MF, Ferreira F, Costa R, , ELRA - European Language Resources Association 2004.

[57] Florian R, Ittycheriah A, Jing H, Zhang T: **Named Entity Recognition through Classifier Combination**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:168–171.

[58] Chieu HL, Ng HT: **Named Entity Recognition with a Maximum Entropy Approach**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:160–163.

[59] Klein D, Smarr J, Nguyen H, Manning CD: **Named Entity Recognition with Character-Level Models**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:180–183.

[60] Zhang T, Johnson D: **A Robust Risk Minimization based Named Entity Recognition System**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:204–207.

[61] Carreras X, Márquez L, Padró L: **A Simple Named Entity Extractor using AdaBoost**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:152–155.

[62] Piskorski J, Sydow M, Kupść A: **Lemmatization of Polish Person Names**. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:27–34, [http://www.aclweb.org/anthology/W/W07/W07-1704].

[63] Erjavec T, Dzeroski S: **Machine Learning of Morphosyntactic Structure: Lemmatizing Unknown Slovene Words**. *Applied Artificial Intelligence* 2004, **18**:17–41.

[64] Taira R, Bui A, Kangarloo H: **Identification of patient name references within medical documents using semantic selectional restrictions**. In *Proceedings of the American Medical Informatics Association 2002*. Edited by Kohane I, Philadelphia, PA, USA: Hanley & Belfus Inc. 2002:757–761.

[65] Thomas SM, Mamlin B, Schadow G, McDonald C: **A Successful Technique for Removing Names in Pathology Reports Using an Augmented Search and Replace Method**. In *Proceedings of the American Medical Informatics Association 2002*. Edited by Kohane I, Philadelphia, PA, USA: Hanley & Belfus Inc. 2002.

[66] Sweeney L: **Replacing personally-identifying information in medical records, the Scrub system**. In *Proceedings of the American Medical Informatics Association 2002*. Edited by Cimino J, Philadelphia, PA, USA: Hanley & Belfus Inc. 1996:333–337.

[67] Ruch P, Baud RH, Rassinoux AM, Bouillon P, Robert G: **Medical Document Anonymization with a Semantic Lexicon**. In *Proceedings of the American Medical Informatics Association 2000* 2000:729–733.

[68] Douglass M, Clifford GD, Reisner A, Moody GB, Mark RG: **Computer-Assisted De-Identification of Free Text in the MIMIC II Database**. *Computers in Cardiology* 2005, **32**:331–334.

[69] D Gupta JG M Saul: **Evaluation of a Deidentification (De-Id) Software Engine to Share Pathology Reports and Clinical Documents for Research**. *American Journal of Clinical Pathology* 2004, **121**(6).

[70] Sibanda T, Uzuner O: **Role of local context in automatic deidentification of ungrammatical, fragmented text**. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics 2006:65–73.

[71] Guillen R: **Automated De-Identification and Categorization of Medical Records**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[72] Aramaki E, Imai T, Miyo K, Ohe K: **Automatic Deidentification by using Sentence Features and Label Consistency**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[73] Wellner B, Huyck M, Mardis S, Aberdeen J, Morgan A, Peshkin L, Yeh A, Hitzeman J, Hirschman L: **Rapidly Retargetable Approaches to De-identification in Medical Records**. *J Am Med Inform Assoc* 2007, **14**(5):564–573, [http://www.jamia.org/cgi/content/abstract/14/5/564].

[74] Hara K: **Applying a SVM Based Chunker and a Text Classifier to the Deid Challenge**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[75] Guo Y, Gaizauskas R, Roberts I, Demetriou G, Hepple M: **Identifying Personal Health Information Using Support Vector Machines**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[76] Grinberg D, Lafferty J, Sleator D: **A Robust Parsing Algorithm For LINK Grammars**. Tech. Rep. CMU-CS-TR-95-125, Carnegie Mellon University, Pittsburgh, PA 1995, [citeseer.ist.psu.edu/grinberg95robust.html].

[77] Clark C, Good K, Jezierny L, Macpherson M, Wilson B, Chajewska U: **Identifying Smokers with a Medical Extraction System**. *J Am Med Inform Assoc* 2008, **15**:36–39, [http://dx.doi.org/10.1197/jamia.M2442].

[78] Aramaki E, Miyo K: **Patient Status Classification by using Rule based Sentence Extraction and BM25-kNN based Classifier**. *Proceedings of i2b2 AMIA workshop* 2006.

[79] Sordo M, Zeng Q: **On Sample Size and Classification Accuracy: A Performance Comparison**. *Lecture Notes in Computer Science* 2005, **3745**:193–201.

[80] Pedersen T: **Determining Smoker Status using Supervised and Unsupervised Learning with Lexical Features**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[81] Carrero FM, Gómez Hidalgo JM, Puertas E, Maña M, Mata J: **Quick Prototyping of High Performance Text Classifiers**. In *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data* 2006.

[82] Cohen AM: **Five-way Smoking Status Classification Using Text Hot-Spot Identification and Error-correcting Output Codes**. *J Am Med Inform Assoc* 2008, **15**:32–35, [http://www.jamia.org/cgi/content/abstract/15/1/32].

[83] Savova GK, Ogren PV, Duffy PH, Buntrock JD, Chute CG: **Mayo Clinic NLP System for Patient Smoking Status Identification**. *J Am Med Inform Assoc* 2008, **15**:25–28, [http://www.jamia.org/cgi/content/abstract/15/1/25].

[84] Heinze DT, Morsch ML, Potter BC, Sheffer J Ronald E: **Medical i2b2 NLP Smoking Challenge: The A-Life System Architecture and Methodology**. *J Am Med Inform Assoc* 2008, **15**:40–43, [http://www.jamia.org/cgi/content/abstract/15/1/40].

[85] Moisio MA: *A Guide to Health Insurance Billing*. Thomson Delmar Learning 2006.

[86] Wiebe J, Wilson T, Bruce RF, Bell M, Martin M: **Learning Subjective Language**. *Computational Linguistics* 2004, **30**(3):277–308.

[87] Riloff E, Wiebe J, Wilson T: **Learning Subjective Nouns using Extraction Pattern Bootstrapping**. In *Proceedings of the Seventh Computational Natural Language Learning Conference*, Edmonton, Canada: Association for Computational Linguistics 2003:25–32, [http://www.aclweb.org/anthology/W/W03/W03-0404].

[88] Miyao Y, Tsujii J: **Probabilistic Disambiguation Models for Wide-Coverage HPSG Parsing**. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan: Association for Computational Linguistics 2005:83–90, [http://www.aclweb.org/anthology/P/P05/P05-1011].

[89] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG: **A simple algorithm for identifying negated findings and diseases in discharge summaries**. *Journal of Biomedical Informatics* 2001, **5**:301–310, [http://citeseer.ist.psu.edu/chapman01simple.html].

[90] **Unified Medical Language System (UMLS)** 2007, [http://www.nlm.nih.gov/research/umls/].

[91] **National Center for Health Statistics - Classification of Diseases, Functioning and Disability** 2007, [http://www.cdc.gov/nchs/icd9.htm].

[92] **ICD9Data.com - Free 2007 ICD-9-CM Medical Coding Database** 2007, [http://www.icd9data.com/].

[93] Goldstein I, Arzumtsyan A, Uzuner O: **Three Approaches to Automatic Assignment of ICD-9-CM Codes to Radiology Reports**. In *Proceedings of the Fall Symposium of the American Medical Informatics Association*, Chicago, Illinois, USA: American Medical Informatics Association 2007:279–283, [http://www.albany.edu/facultyresearch/clip/papers/AMIA-2007.pdf].

[94] Larkey LS, Croft WB: **Technical Report - Automatic assignment of icd9 codes to discharge summaries**. *PhD thesis*, University of Massachusetts at Amherst, Amherst, MA 1995.

[95] Lussier Y, Shagina L, C F: **Automated ICD-9 encoding using medical language processing: a feasibility study**. In *Proceedings of AMIA Symposium 2000* 2000:1072.

[96] de Lima LRS, Laender AHF, Ribeiro-Neto BA: **A hierarchical approach to the automatic categorization of medical documents**. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, New York, NY, USA: ACM Press 1998:132–139.

[97] **International Challenge: Classifying Clinical Free Text Using Natural Language Processing** 2007, [http://www.computationalmedicine.org/challenge/index.php].

[98] Patrick J, Zhang Y, Wang Y: **Developing Feature Types for Classifying Clinical Notes**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:191–192, [http://www.aclweb.org/anthology/W/W07/W07-1027].

[99] Aronson AR, Bodenreider O, Demner-Fushman D, Fung KW, Lee VK, Mork JG, Neveol A, Peters L, Rogers WJ: **From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:105–112, [http://www.aclweb.org/anthology/W/W07/W07-1014].

[100] Crammer K, Dredze M, Ganchev K, Pratim Talukdar P, Carroll S: **Automatic Code Assignment to Medical Text**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:129–136, [http://www.aclweb.org/anthology/W/W07/W07-1017].

[101] Aizerman A, Braverman EM, Rozoner LI: **Theoretical foundations of the potential function method in pattern recognition learning**. *Automation and Remote Control* 1964, **25**:821–837.

[102] Boser BE, Guyon I, Vapnik V: **A Training Algorithm for Optimal Margin Classifiers**. In *Computational Learing Theory* 1992:144–152, [citeseer.ist.psu.edu/boser92training.html].

[103] Zhou X, Zhang X, Hu X: **Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining**. *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, in press.

[104] Witten IH, Frank E: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, second edition 2005, [http://www.amazon.fr/exec/obidos/ASIN/0120884070/citeulike04-21].

[105] McCallum AK: **MALLET: A Machine Learning for Language Toolkit** 2002. [Http://mallet.cs.umass.edu].

[106] ACE: **Annotation Guidelines for Entity Detection and Tracking** 2004, [http://projects.ldc.upenn.edu/ace/docs/EnglishEDTV4-2-6.pdf].