Machine Learning techniques for applied Information Extraction

Richárd Farkas

Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences
and the University of Szeged

June 2009

A thesis submitted for the degree of doctor of philosophy
of the University of Szeged, Faculty of Science and Informatics

University of Szeged, Faculty of Science and Informatics
Doctoral School of Computer Science

Preface

Information Extraction is the field for automatically extracting useful information from natural language texts. There are several applications which require Information Extraction like monitoring companies (extracting information from the news about them), the transfer of patient data from discharge summaries to a database and building biological knowledgebases (e.g. gathering protein interaction pairs from the scientific literature).

The early Information Extraction solutions applied manually constructed expert rules which required both domain and linguist/decision system experts. On the other hand, the *Machine Learning* approach requires just examples from the domain expert and builds the decision rules automatically. Natural Language Processing tasks can be formulated as classification tasks in Machine Learning terminology and this means that they can be solved by statistical systems that discover patterns and regularities in the labeled set of examples and exploit this knowledge to process new documents.

The main aim of this thesis is to examine various Machine Learning methods and discuss their suitability in real-world Information Extraction tasks. Among the Machine Learning tools, several less frequently used ones and novel ideas will be experimentally investigated and discussed. The tasks themselves cover a wide range of tasks from language-independent and multi-domain Named Entity recognition (word sequence labelling) to Name Normalisation and Opinion Mining.

Richárd Farkas, June 2009.

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 of the Hungarian Academy of Sciences.

I am indebted to my senior colleagues who showed me interesting undiscovered fields and helped give birth to new ideas during our inspirable discussions. In alphabetical order: László Balázs, András György, Márk Jelasity, Gabriella Kókai and Csaba Szepesvári.

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

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 to 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 my wife Kriszta for her endless love, support and inspiration. Last, but not least I wish to thank my parents, my sisters and brothers 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

Chapter 1

Introduction

"Everything should be made as simple as possible, but no simpler."
Albert Einstein

1.1 Problem definition

Due to the rapid growth of the Internet and the birth of huge multinational companies, the amount of publicly available and in-house information is growing at an incredible rate. The greater part of this is in textual form which is written by humans for human reading. The manual processing of these kinds of documents requires an enormous effort.

Information Extraction is the field for extracting useful information from natural language texts and providing a machine-readable output. There are several applications which require Information Extraction like monitoring companies (extracting information from the news about them), the transfer of patient data from discharge summaries to a database and building biological knowledgebases (e.g. gathering protein interaction pairs from the scientific literature).

The early Information Extraction solutions applied manually constructed expert rules which required both domain and linguist/decision system experts. On the other hand, the *Machine Learning* approach requires just examples from the domain expert and builds the decision rules automatically. The aim of this thesis is to examine various Machine Learning methods and discuss their suitability in real-world Information Extraction tasks. Among the Machine Learning tools several less frequently used ones and novel ideas will be experimentally investigated and discussed.

1.2 Contributions

In this thesis several practical Information Extraction applications will be presented which were developed together with colleagues. The applications themselves cover a wide range of different tasks from entity recognition (word sequence labelling) to word sense disambiguation, various domains from business news texts to medical records and biological scientific papers. We will demonstrate that a task-specific choice of Machine Learning tools and several simple considerations can result in a significant improvement in the overall performance. Moreover, for specific text mining tasks, it is feasible to construct systems that are useful in practice and can even compete with humans in processing the majority of textual data.

A reference corpus (labeled textual dataset) is available for each addressed task. These corpora provide the opportunity for the objective evaluation of automatic systems and we shall use them to experimentally evaluate our Machine Learning-based systems. So-called shared tasks are often organised based on these corpora which attempt to compare solutions from all over the world. The task-specific systems developed at the University of Szeged achieved good rankings in these challenges.

Several tasks were solved for Hungarian and English texts at the same time. We will show that – however Hungarian has very special characteristics (e.g. agglutination and free word order) – at this level of processing (i.e. having the output of language-specific modules like morphological analysis) the same Machine Learning systems can perform well on the two languages.

From a Machine Learning point of view, we will consider solutions for exploiting external resources (introduced in Part II) as valuable and novel results. We think that this field will be more intensively studied in the near future.

1.3 Dissertation roadmap

Here we will summarise our findings for each chapter of the thesis and present the connection between the publications of the author referred to in the thesis and the results described in different chapters in a table.

This thesis is comprised of two main parts. The first part (chapters 3-5) deals with supervised learning approaches for Information Extraction, while the second (chapters 6, 7 and 8) discusses several strategies for exploiting external resources.

The second, introductory chapter provides the necessary background definitions and briefly introduces the topics, tasks and datasets for each problem addressed in the thesis. The third chapter presents supervised Machine Learning approaches for *Named Entity Recognition* and *Metonymy Resolution* tasks along with experimental results and comparative, task-oriented discussions about the models employed.

The fourth chapter describes several model-combination techniques including the author's *Feature set split and recombination* approach. Based on these approaches we developed a complex Named Entity Recognition system which is competitive to the published state-of-the-art systems while has a different theoretical background compared to the widely used models.

The last chapter of the first part deals with adaptation issues of the supervised systems. Here, the *adaptation* of our Named Entity Recognition system between languages (from Hungarian to English) and domains (from newswire to clinical) will be discussed.

The Part II of the thesis introduces several approaches which go beyond the usual supervised learning environment. The sixth chapter is concerned with semi-supervised learning. Here, the basic idea is to exploit unlabeled data in order to build better models on labeled data. Several novel approaches will be introduced which use the WWW as an unlabeled corpus. We will show that these techniques efficiently solve the *Named Entity lemmatisation* problem and achieve remarkable results in Named Entity Recognition as well.

The seventh chapter focuses on the integration possibilities of machine learnt models and existing knowledge sources/expert systems. Clinical knowledgebases are publicly available in a great amount and two *clinical Information Extraction* systems which seek to extract diseases from textual discharge summaries will be introduced here.

In the last chapter of the thesis we introduce two applied Information Extraction tasks which are handled by exploiting non-textual inter-document relations and textual cues in parallel. We constructed and used the *inverse co-authorship graph* for the *Gene Name Disambiguation* task and the *response graph* was utilised for the *Opinion Mining* task.

			Chapter					
			3	4	5	6	7	8
ACTA	2006	[1]	•	•				
LREC	2006	[2]	•					
SEMEVAL	2007	[3]	•					
DS	2006	[4]		•				
JAMIA	2007	[5]			•			
BIONLP	2008	[6]			•			
ICDM	2007	[7]				•		
TSD	2008	[8]				•		
BMC	2007	[9]					•	
JAMIA	2009	[10]					•	
BMC	2008	[11]						•
DMIIP	2008	[12]						•

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

Table 1.1 summarises the relationship among the thesis chapters and the more important[1] referred publications of the author. Here we list the most important results in each paper that are regarded as the author's own contributions. We should 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 [11], which describes only the author's own results. [2] has been omitted from the list as all the results described in this paper are counted as shared contributions of the authors. For [3], the author only made marginal contributions.

- ACTA 2006 [1]
  - Comparison of supervised learners.
  - Stacking approach.
- SEMEVAL 2007 [3]
  - Comparative analysis of C4.5 and Logistic Regression.
- DS 2006 [4]
  - The architecture of the complex Named Entity Recognition model
  - The feature set split and recombination method.
  - Boosting experiments.
  - Post-processing rules.
- JAMIA 2007 [5]
  - Trigger-based bagging method.
  - The standardisation phase.
- BIONLP 2008 [6]
  - Statistical investigations for differences between the medical and biological domains.
- ICDM 2007 [7]
  - Using web frequencies for phrase boundary extension.
  - The most frequent rule heuristic.
- TSD 2008 [8]
  - Feature set construction and transformations for NE lemmatisation and separation.

[1] For a full list of publications, please visit http://www.inf.u-szeged.hu/~rfarkas/publications.html.

  - All of the Machine Learning experiments.

- BMC 2007 [9]
  - Combination strategies for expert rules and Machine Learning methods.
  - Negation and speculation-based language pre-processing.
  - Multi-label classification approaches.
  - All of the data-driven experiments.

- JAMIA 2009 [10]
  - Statistical methods for term identification.
  - Statistical methods for context detection.

- BMC 2008 [11]
  - All of the results in the paper.

- DMIIP 2008 [12]
  - The general idea of using response graphs for Opinion Mining.

Chapter 2

Background

In this chapter we will provide the background to understanding key concepts in the thesis. First we will give a general overview of the Information Extraction and Named Entity Recognition problems, then we will define basic common notations and definitions for Machine Learning. Finally, we will describe each Information Extraction tasks and datasets used in the experiments presented in the thesis.

2.1 Human Language Technology

Due to the rapid growth of the Internet and the globalisation process the amount of available information is growing at an incredible rate. The greater part of the new data sources is in textual form (e.g. every web page contains textual information) that is intended for human readers, written in a local natural language. This amount of information requires the involvement of the computer into the processing tasks. The automatic or semi-automatic processing of raw texts requires special techniques. *Human Language Technology* (HLT) is the field which deals with the understanding and generation of natural languages (i.e. languages written by humans) by the computer. It is also known as *Natural Language Processing* or *Computational Linguistics*.

The computerised full understanding of natural languages is an extremely hard (or even impossible) problem. Take, for example, the fact that the meaning of a written text depends on the common background knowledge of the author and the targeted audience, the cultural environment (e.g. use of idioms), the conditions of the disclosure and so on. The state-of-the-art techniques of Human Language Technology deals with the identification of syntactic and semantic clues in the text. *Syntactics* is concerned with the formal relations between expressions i.e. words, phrases and sentences while *semantics* is concerned with relations between expressions and what they refer to and it is deals with the relationship among meanings.

The chief application fields of HLT are:

Information Retrieval searches for the most relevant documents for a query (like Google).

Information Extraction goes below the document level, analysis texts in depth and returns with the targeted exact information.

Machine Translation seeks to automatically translate entire documents from one natural language to another (e.g. from English to Hungarian).

Summarization is the creation of a shortened version of a text so that it still contains the most important points of the original text.

This thesis is concerned with investigating and determining of the necessary and useful tools for applied Information Extraction tasks.

2.1.1 The Information Extraction Problem

The goal of *Information Extraction* (IE) is to automatically extract structured information from unstructured, textual documents like Web pages, corporate memos, news articles, research reports, e-mails, blogs and so on. The output is structured information which is categorized and semantically well-defined data, usually in a form of a relational database.

In contrast to Information Retrieval (IR) whose input is a set of documents and the output are several documents that must read by the user, IE works on a few documents and returns detailed information. These information are useful and readable for humans like browsing, searching, sorting (e.g. in Excel sheets) and for the computer as well, thus IE is often the preprocessing step in a Data Mining system (which processes structural data) [13].

Example applications include the gathering of data about companies, corporate mergers from the Web or protein-protein interactions from biological publications. For example, from the sentence "*Eric Schmidt joined Google as chairman and chief executive officer in 2001.*" we can extract the information tuple:

{COMPANY=*Google*, CEO=*Eric Schmidt*, CEO_START_DATE=*2001*}

The first domain of application was the newswire one (especially business news) at the end of the '80s at the Message Understanding Conferences (MUC) [14]. With the dramatic growth of the Internet, web pages have become the focus on points such as personal information (job titles, employment histories, and educational backgrounds are semi-automatically collected at ZoomInfo[1]). Recently, the main application domain

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

has become the information extraction from biological scientific publications [15] and medical records [16].

Similar to domains, the target languages of IE has also grown in the past fifteen years. Next to English the most investigated languages are Arabic, Chinese and Spanish (Automatic Content Extraction evaluations [17]), but smaller languages like Hungarian (e.g. [18]) are the subject of research as well.

The Information Extraction task can be divided into several subtasks. These are usually the following:

Named Entity Recognition is the identification of mentions of entities in the text.

Co-reference resolution collects each mention – including pronouns, bridging references and appositions – which refers to a certain entity.

Relation Detection seeks to uncover relations between entities, such as father-son or shop-address relation.

Identification of roles of the entities in an event usually means filling in slots of a pre-defined event frame.

2.1.2 Named Entity Recognition

A *Named Entity* (NE) is a phrase in the text which uniquely refers to an entity of the world. It includes proper nouns, dates, identification numbers, phone numbers, e-mail addresses and so on. As the identification of dates and other simpler categories are usually carried out by hand-written regular expressions we will focus on proper names like organisations, persons, locations, genes or proteins.

The identification and classification of proper nouns in plain text is of key importance in numerous natural language processing applications. It is the first step of an IE system as proper names generally carry important information about the text itself, and thus are targets for extraction. Moreover *Named Entity Recognition* (NER) can be a stand-alone application as well (see Section 2.3.4) and besides IE, Machine Translation also has to handle proper nouns and other sort of words in a different way due to the specific translation rules that apply to them.

The NER problem has two levels. First the expressions in the text must be identified then the semantic class of the entity must be chosen from a pre-defined set. As the identification is mainly based on syntactic clues and classification requires semantic disambiguation, the NER task lies somewhere between syntactic and semantic analysis. The classification of NEs is a hard problem because (i) the NE classes are open, i.e. there will never be a list which consists of each person or organisation names and (ii) the context of the phrase must be investigated, e.g. the expression *Ford* can refer to the company, the car itself or to *Henry Ford*.

The most studied entity types are three specializations of proper names: names of persons, locations and organizations. Sekine et al. [19] defined a Named Entity hierarchy with about 200 categories, which includes many fine-grained subclasses, such as international organization, river, or airport, and also has a wide range of categories, such as product, event, substance, animal or religion. Recent interest in the human sciences has led to researchers dealing with new entity types such as gene, protein, DNA, cell line and drug, chemical names.

2.2 Machine Learning

The first approaches for Information Extraction tasks were based on hand-crafted expert rules. The construction and maintenance of such a rule system is very expensive and requires a decision system specialist. Machine Learning techniques construct decision rules automatically based on a training dataset – a manually annotated corpus in Information Extraction. The cost of the training set's construction is less than the cost of a hand-written rule set because the former one requires just domain knowledge i.e. labelling examples instead of decision system engineering. This thesis is concerned with the investigation of Machine Learning tools for Information Extraction tasks and their application in particular domains.

2.2.1 Basic concepts

Machine Learning is a broad subfield of Artificial Intelligence. The learning task of the machine in this context is the automatic detection of certain *pattern*s and *regularities*. It operates on large datasets which are intractable to humans but contain useful information. The statistically identified patterns help humans understand the structure of the underlying problem and they can be used to make predictions.

A machine learns with respect to a particular *task*. In general, a task consists of N number of *object*s (also called as *instance* or *entity*) $x_{1..N}$ and a *performance metric* v. Then the goal is to detect patterns which *model* the underlying data and can make predictions whose quality is decided by the performance metric. For example, a task might be the automatic identification of spam (spam detection). Here the objects are e-mails and the performance metric could be the accuracy i.e. the ratio of correctly-identified e-mails on an unseen test set. Note that detecting patterns on two different sets of e-mails are two different tasks.

There are two main types of Machine Learning tasks. With *classification* the learned model has to choose from a pre-defined set of classes, while *regression* forecasts a real value within a certain interval for a given test instance. In this thesis we shall just deal with classification as Human Language Technology usually does. The true class of an

instance i shall be denoted by $c_i \in C$ and the predicted class by $\hat{c}_i \in C$. Let us suppose that there exists a test (or evaluation) set for each task and that the performance of a classification system is measured on this set by $v(c, \hat{c}) : C^N \times C^N \to \Re$, where c and $\hat{c}$ are the sets of etalon and predicted labels on the whole evaluation dataset.

The objects can be characterized by a set of *feature*s (also known as *attribute*s). We will denote the feature space by F and the jth feature's value of the ith instance by x_{ij}. A classification model $f : F \to C$ statistically detects some relationship among the features and the class value. The value range of a feature can be *numeric* (real or integer), *string* or *nominal*. With the latest, the possible values of a certain feature arise from a finite set and there is no distance metric or ordering among the elements. In the spam detection example, the length of the e-mail is a numeric feature, the first word of the e-mail is a string and whether the e-mail contains a particular word is a nominal feature.

2.2.2 Overfitting and generalisation

In inductive learning the goal is to build a classification or regression model on the training data in order to classify/forecast previously unseen instances. It means that the trained model must find the golden mean between fitting the training data and generalising on the test (unseen) data. The trade-off between them can be achieved by varying the complexity of the learning method (which in most methods can be regularised by changing the values of a few parameters). As the method becomes increasingly complex, it will became able to capture more complicated underlying structure of a dataset but its generalisation capability decreases as the model is *overfit* to the training data.

The *generalisation error* (or *test error*) is the expected error over the test sample. It cannot be estimated from the error calculated on the training set (*training error*) because if the training error decreases (i.e. the model complexity grows) its generalisation may become poorer. The most common approach to estimate the test error and thus selecting the best model or model parameters is separating a *development dataset* (or *validation set*) from the training set. Then the models can be trained on the remaining training set and evaluated on the development set. This separation may be random and may be repeated with the averaging the errors measured (this is called *cross-validation*). This approach preserves the inviolateness of the test data [20].

2.2.3 Supervision in Machine Learning

The level of supervision in Machine Learning refers to the availability of manually labeled instances (i.e. their classes are assigned by humans). In *supervised learning*, we use just labeled training samples. However there usually exist a huge amount of unlabeled instances and useful patterns can be extracted from them. A *semi-supervised learning*

task labeled X_L and unlabeled X_U samples are used together. Lastly, *unsupervised learning* works on just unlabeled data and the aim is to find regularities or patterns in the input (the most common known unsupervised task is clustering). Even so, every experimental Machine Learning setting should contain a labeled test set to be able to provide an objective evaluation of the methods applied.

As several semi-supervised approaches will be described in this thesis, we should present a deeper categorisation of them. Semi-supervised approaches can be categorised according to the handling strategy of test instances. *Inductive method*s construct models which seek to make a correct prediction on every unseen test instance in general, whereas *transductive method*s optimise just for one particular evaluation set, i.e. they have to be re-trained for each new test set [21].

2.2.4 Machine Learning in Information Extraction

Information Extraction tasks have several special properties from a Machine Learning point of view:

- An IE task is usually built up as a chain of several classification subtasks (see Section 2.1.1) and the objects of each classification tasks are words or phrases.

- The words of a text are not independent of each other. During the feature space construction, the training and evaluation sequences of words have to be taken into account. IE tasks can usually be understood as a *sequence labeling* problem, so the evaluation and optimization are performed on sequences (e.g. sentences) instead of words.

- The majority of the feature set is nominal in IE tasks. This fact lone casts doubt on the suitablity of popular numeric learners (e.g. SVM) for tasks like these.

- The Internet abounds in textual data which can be considered as an unlabeled corpus for general HLT tasks. Generally speaking, there usually exists a large amount of domain texts for a certain IE task in a cheap and natural form.

2.3 Tasks and corpora used

In this section we shall introduce seven IE tasks and the corresponding corpora which were used when we conducted our experiments. In the case of the Hungarian NE corpus we will also describe the annotation process in detail as the author participated in the building of this corpus.

2.3.1 English Named Entity corpus

The identification of Named Entities can be regarded as a tagging problem where the aim is to assign the correct label to each token in a raw text. This classification determines whether the lexical unit in question is part of a NE phrase and if it is, which category it belongs to. The most widely used NE corpus, the CoNLL-2003 corpus [22], follows this principle and assigns a label to each token. It includes annotations of *person*, *location*, *organization names* and *miscellaneous entities*, which are proper names but do not belong to the three other classes.

The CoNLL-2003 corpus and the corresponding shared task can be regarded as the worldwide reference NER task. It is a sub-corpus of the Reuters Corpus[2], consisting of newswire articles from 1996 provided by Reuters Inc. The data is available free of charge for research purposes and contains texts from diverse domains ranging from sports news to politics and the economy. The corpus contains some linguistic preprocessing as well. On all of this data, a tokeniser, part-of-speech tagger, and a chunker (the memory-based MBT tagger [23]) were applied. A NER shared task was performed on the CoNLL-2003 corpus, which provided a new boost to NER research in 2003. The organisers of the shared task divided the document set into a training set, a development set and a test set. Their sizes are shown in the table below.

	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617

Table 2.1: The size and the label distribution of the CoNLL-2003 corpus.

2.3.2 Hungarian Named Entity corpus

The Named Entity Corpus for Hungarian is a sub corpus of the Szeged Treebank [24][3], which contains 1.2 million words with tokenisation and full morphological and syntactic annotation was 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 (see the previous section). The corpus is available free of charge for research purposes[4].

Short business news articles collected from MTI (Hungarian News Agency[5]) consti-

[2] http://www.reuters.com/researchandstandards/

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

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

[5] www.mti.hu

tute a part of the Szeged Treebank, 225,963 words in size, covering 38 topics related to the NewsML topic coding standard, ranging from acquisition to stock market changes to new plant openings. Part of speech codes generated automatically by a POS tagger [25] developed at the University of Szeged were also added to the database. In addition we provided some gazetteer resources in Hungarian (Hungarian first names, company types, list of the names of countries, cities, geographical name types and a stopword list) that we used for experiments to build a Machine Learning-based model.

The dataset has some interesting aspects relating to the distribution of class labels (see Table 2.2) which is induced by the domain specificity of the texts. Organization class – which turned out to be harder to recognize than, for example, person names – has a higher frequency in this corpus than in other standard corpora for other languages.

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

Table 2.2: Label distribution in the Hungarian NER corpus.

We divided the corpus into 3 parts, namely a training, a development set and a test subcorpus, following the protocol of the CoNLL-2003 NER shared 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 2.3: The size of the Hungarian NER datasets.

2.3.3 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 a NE recognizer and classifier

systems for Hungarian. To guarantee the precision of tagging we created an annotation procedure with three stages [2].

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 the words that got 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 each occurrence of each shopping mall name, regardless whether the actual occurrence caused a disagreement or not. We did this so as to ensure the consistency of the annotation procedure. The resulting corpus after the final, third stage of consistency checking was considered error-free.

Creating error-free resources of a reasonable size has a very high cost and, in addition, publicly available NE tagged corpora 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.

2.3.4 Anonymisation of medical records

The process of removing personal health information (PHI) from clinical records is called de-identification. This task is crucial in the human life sciences because a de-identified text can be made publicly available for non-hospital researchers as well, to facilitate research on human diseases. However, the records about the patients include explicit personal health information, and this fact hinders the release of many useful

datasets because their release would jeopardise individual patient rights. According to the guidelines of Health Information Portability and Accountability Act (HIPAA) the medical discharge summaries released must be free of the following seventeen categories of textual PHI: first and last names of patients, their health proxies, and family members; doctors' first and last names; identification numbers; telephone, fax, and pager numbers; hospital names; geographic locations; and dates. Removing these kinds of PHI is the main goal of the de-identification process.

We used the de-identification corpus prepared by researchers of the I2B2 consortium[6] for the de-identification challenge of the 1st I2B2 Workshop on Natural Language Processing Challenges for Clinical Records [26]. 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. Since the challenge organisers wanted to concentrate on the separation of PHI and non-PHI tokens, they made the dataset more challenging with two 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.

Table 2.4 lists the size and label distribution of the train and test sets used on the I2B2 shared task.

2.3.5 Metonymy Resolution

In linguistics metonymy means using one term, or one specific sense of a term, to refer to another, related term or sense. Metonymic usage of NEs is frequent in natural language. For example in the following example *Vietnam*, the name of a location, refers to an event (the war) that happened there [27]:

Sex, drugs, and Vietnam have haunted Bill Clinton's campaign.

In order to support automatic distinction among the metonymic senses of the NEs – which can be regarded as a more fine-grained NE classification task – Markert et al.

[6] www.i2b2.org

	Train Set		Test Set	
	Tokens	Phrases	Tokens	Phrases
Non-PHI	310504	–	133623	–
Patients	1335	684	402	245
Doctors	5600	2681	2097	1070
Locations	302	144	216	119
Hospitals	3602	1724	1602	676
Dates	5490	5167	2161	1931
IDs	3912	3666	1198	1143
Phone Numbers	201	174	70	58
Ages	13	13	3	3

Table 2.4: The label distribution of the de-identification dataset.

[28] constructed a corpus which focuses on the metonymic categories of *organisation*s and *location*s. The corpus consists of four sentence long contexts around the target NEs from the British National Corpus [29] and the aim was to classify each target NE to one of the metonymy categories. Table 2.5 shows the size of the corpus, the train/test split and its label distribution. This corpus was the evaluation base of the Metonymy Resolution shared task at Semeval-2007 [28] where with his colleagues he achieved excellent results.

LOCATION

class	train	test
literal	737	721
mixed	15	20
othermet	9	11
obj-for-name	0	4
obj-for-representation	0	0
place-for-people	161	141
place-for-event	3	10
place-for-product	0	1
total	925	908

ORGANISATION

class	train	test
literal	690	520
mixed	59	60
othermet	14	8
obj-for-name	8	6
obj-for-representation	1	0
org-for-members	220	161
org-for-event	2	1
org-for-product	74	67
org-for-facility	15	16
org-for-index	7	3
total	1090	842

Table 2.5: The label distribution of the metonymy corpus.

2.3.6 ICD-9-CM coding of medical records

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].

There are official guidelines for coding radiology reports [31]. These guidelines define the codes for each disease, symptom and also place limitations 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. The manual coding of medical records is prone to errors as human annotators have to consider thousands of possible codes when assigning the right ICD-9-CM labels to a document.

Automating the assignment of ICD-9-CM codes for radiology records was the subject of a shared task challenge organized by the Computational Medicine Center (CMC) in Cincinatti, Ohio in the spring of 2007. The detailed description of the task, and the challenge itself, can be found in [16], and also online[7].

We used the datasets made available by the shared task organisers to train and evaluate our automatic ICD coder system. The radiology reports of this dataset had two parts *clinical history* and *impression* and had typically 4-5 sentences of lengths. The gold standard of the dataset was the majority annotation of three human annotators who could assign as many codes as they liked (*multi-labeling problem*). After a few cleaning steps [16], the majority annotation consisted of 45 distinct ICD-9-CM codes in 94 different code-combination. There were a few frequent codes but the half of the codes had less than 15 occurrences (see Figure 2.1). The whole document set was divided into two parts, a training set with 978 documents and a testing set with 976 records.

2.3.7 Gene Symbol Disambiguation

The goal of Gene Name Normalisation (GN) [32] is to assign a unique identifier to each gene name found in a text. However due to the diversity of the biological literature,

[7] http://www.computationalmedicine.org/challenge/

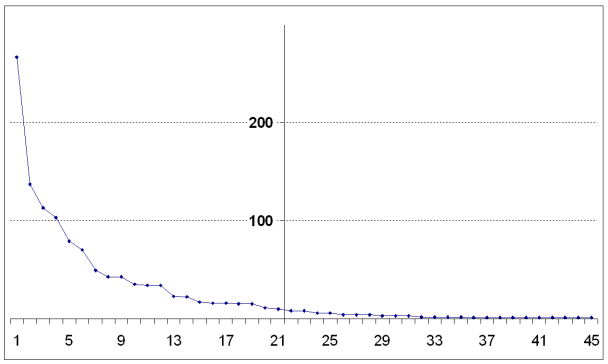

Figure 2.1: The frequencies of the 45 ICD labels in descending order.

one name can refer to different entities. The task of Gene Symbol Disambiguation (GSD) [33] is to choose the correct sense (gene referred by unique identifier) based on the contexts of the mention in the biological article.

We will present experimental results on the GSD datasets built by Xu et al. [34, 35]. In [34] Xu and his colleagues took the words of the abstracts, the MeSH codes provided along with the MedLine articles, the words of the texts and some computer tagged information (UMLS CUIs and biomedical entities) as features while in [35] they experimented with the use of combinations of these features. They used them to get manually disambiguated instances (training data).

The GSD datasets for yeast, fly and mouse are generated using MedLine abstracts and the Entrez 'gene2pubmed' file [36], which is manually disambiguated [34]. The dataset for human genes was derived [35] from the training and evaluation sets of the BioCreative II GN task [37]. The most important statistics of these evaluation sets are listed in Table 2.6.

Organism	#test	Avg. #senses	Avg. train size	Avg. #synonyms avail.
Human	124	2.35	122.09	12.36
Mouse	7844	2.33	263.00	5.36
Fly	1320	2.79	35.69	9.51
Yeast	269	2.08	11.00	2.32

Table 2.6: The characteristics of the evaluation sets used.

Part I

Supervised learning for Information Extraction tasks

Chapter 3

Supervised models for Information Extraction

The standard approaches for solving IE tasks work in a supervised setting where the aim is to build a Machine Learning model on training instances. In this chapter, the most common approaches will be presented along with empirical comparative results and a discussion.

3.1 Token-level classification

In our first investigations of Machine Learning models we carried out experiments on the Hungarian NE dataset (see Section 2.3.2). We viewed the identification of named entities as a classification problem where the aim is to assign the correct tag (label) for each token in a plain text. This classification determines whether the lexical unit in question is part of a proper noun phrase and, if it is, which category it belongs to. We made a comparison study on four well-known classifiers (C4.5 decision tree, Artificial Neural Networks, Support Vector Machines and Logistic Regression) which are based on different theoretical backgrounds [1].

3.1.1 Decision trees

A *decision tree* is a tree whose internal nodes represent a decision rule, its descendants are the possible outcomes of the decision and the leaves are the final decisions (class labels in the classification case). *C4.5* [38] – which is based on the well-known *ID3* tree learning algorithm [39] – is able to learn a decision tree with discrete classes from labeled examples. The result of the learning process is an axis-parallel decision tree. During the training, the sample space is divided into subspaces by hyperplanes that are parallel to every axis but one. In this way, we get many n-dimensional rectangular regions that are labeled with class labels and organized in a hierarchical way, which

can then be encoded into a tree. Since C4.5 considers attribute vectors as points in an n-dimensional space, using continuous sample attributes naturally makes sense. For knowledge representation, decision trees use the "divide and conquer" technique, meaning that regions (which are represented by a node of the tree) are split during learning whenever they are insufficiently homogeneous. C4.5 executes a split at each step by selecting the most inhomogeneous feature. The measure of homogeneity is usually the so-called GainRatio:

$$GainRatio(x_i, S)\frac{H(S) - \sum_{v \in x_i} \frac{|S_v|}{|S|} H(S_v)}{\sum_{v \in x_i} \frac{|S_v|}{|S|} \log \frac{|S_v|}{|S|}}, \tag{3.1}$$

where x_i is the feature in question, S is the set of entities belonging to the node of the tree, S_v is the set of objects whit $x_i = v$ and $H(S)$ is the Shannon entropy of class labels on the set S. One great advantage of the method its training and testing time complexity; in the average case it is $O(|F|n \log n) + O(n \log^2 n)$ and $O(\log n)$, where $|F|$ is the number of features and n is the number of samples [40]. As $|F|$ (several hounders in the compact case and 10^4 otherwise) is higher than $\log n$ in IE tasks, the training time complexity can be simplified to $O(|F|n \log n)$. This makes the C4.5 algorithm suitable for performing preliminary investigations.

To avoid the overfitting of decision trees, several pruning techniques have been introduced. These techniques include heuristics for regulating the depth of the tree by constraints on splitting. We used the J48 implementation of the WEKA package [40], which regulates pruning by two parameters. The first gives a lower bound for the instances on each leaves, while the second one defines a confidence factor for the splits.

3.1.2 Artificial Neural Networks

Artificial Neural Networks (ANN) [41] were inspired by the functionality model of the brain. They consist of multiple layers of interconnected computational units called neurons. The most well-known ANN is a feed-forward one, where each neuron in one layer has directed connections to the neurons of the subsequent layer (*Multi Layer Perceptron*). A neuron has several inputs which can come from the surrounding region or from other perceptrons. The output of the neuron is its activation state, which is computed from the inputs using an activation function $a()$. The hidden (or inner) layers of the networks usually apply the sigmoid function as the activation function, so

$$a(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$

In this framework, training means finding the optimal $\mathbf{w}$ weights according to the training dataset and a pre-defined error function. The Multi Layer Perceptrons can be trained by a variety of learning techniques, the most popular being the *back-propagation*

one. In this approach the error is fed back through the layers of the network. Using this information the learning algorithm adjusts the **w** of each neuron in order to reduce the value of the error function by some small amount (the weights are initially assigned small random values). The method of gradient descent is usually applied for this adjustment of the weights, where the derivative of the error function with respect to the network weights is calculated and the weights are then modified so that the error decreases. For this reason, back-propagation can only be applied on networks with differentiable activation functions. Repeating this process for a sufficiently large number of training epochs, the network usually converges to a state where the computed error is small.

3.1.3 Support Vector Machines

The well-known and widely used Support Vector Machines (SVMs) [42] is a discriminative learning algorithm. It separates data points of different classes with the help of a hyperplane in the transformed space. The created separating hyperplane has a margin of maximal size with a proved optimal generalization capacity. There exists several SVM formalism the best known being C-SVM. The optimisation problem in C-SVM – which is the most widely used formalism – becomes:

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_i \xi_i$$

$$y(\mathbf{w}^T\phi(x) + b) > 1 + \xi_i \ , \ \xi > 0,$$

where ϕ is the transformation function and C is the regularisation parameter which can help to avoid overfitting.

This quadratic programming optimization problem is solved in its dual form. SVMs apply the "kernel-idea" [43], which is simply a proper redefinition of the two-operand operation of the dot product $K(x, y) = \phi(x)^T\phi(y)$. We can have an algorithm that will now be executed in a different dot product space, and is probably more suitable for solving the original problem. Of course, when replacing the operand, we have to satisfy certain criteria, as not every function is suitable for implicitly generating a dot product space. The family of Mercer kernels is a good choice (based on the Mercer's theorem) [44].

The key to the success of SVM in an application is based on the appropriate choice of the kernel. In our experiments we tried out several kernels and the discrete version of the polynomial kernel $(\gamma x^T y + r)^3$ proved to be the best one. An important feature of margin maximization is that the calculation of the hyperplane is independent of the distribution of the sample points.

3.1.4 The NER feature set

We employed a very rich feature set (which was partly based on the model described in [45]) for our word-level classification model, describing the characteristics of the word itself along with its actual context (a moving window of size four). We were interested in the behaviour of the various learning algorithms so we used the same feature set for each. Our features fell into the following main categories:

orthographical features: capitalization, word length, common bit information about the word form (contains a digit or not, has uppercase character inside the word, and so on). We collected the most characteristic character-level bi/trigrams from the train texts assigned to each NE class,

gazetteers of unambiguous NEs from the train data: we used the NE phrases which occurred more than five times in the train texts and got the same label in more than 90% of the cases,

dictionaries of first names, company types, denominators of locations,

frequency information: frequency of the token, the ratio of the token's capitalized and lowercase occurrences, the ratio of capitalized and sentence beginning frequencies of the token which was derived from the Szoszablya webcorpus [46],

phrasal information: chunk codes and the forecasted class of a few preceding words (we carried out an online evaluation),

contextual information: automatic POS codes, sentence position, trigger words (the most frequent and unambiguous tokens in a window around the NEs) from the train text, the word between quotes, and so on.

Here we used a *compact feature representation* approach. By compact representation we mean that we gathered together similar features in lists. For example, we did not have thousands of binary features for each Hungarian town names, but we performed a preprocessing step where the most important names were filtered, so we just used one feature whether the filtered list contained the token in question. We applied this approach on the gazetteers, dictionaries and all the lists gathered from the train texts which resulted in a feature set of tractable size (184 attributes).

In many approaches presented in the literature the starting tokens of Named Entities are distinguished from the inner parts of the phrase [22] when the entity phrases directly follow each other. This turns out to be useful when several proper nouns of the same type follow each other,as it makes it possible for the system to separate them instead of treating them as one entity. When doing so, one of the "I-", "B-" (for inside and begin) labels also has to be assigned to each term that belongs to one of the four classes

used. In our experiments we decided not to do this for two reasons. First, for some of the classes we barely had enough examples in our dataset to separate them well, and it would have made the data available even sparser. Second, in Hungarian texts, proper names following each other are almost always separated by punctuation marks or a stopword. There are several approaches [47][48] which distinguish every phrase start (not just for phrases without separation) but they do not report any significant improvements. This is due to the doubled number of predictable class labels, which seems to be intractable with this size of training examples.

3.1.5 Comparison of supervised models

The standard evaluation metric of the NER systems is the phrase-level F-measure which was introduced at the CoNLL [22] conferences[1]. In this case we calculated the precision, recall and $F_{\beta=1}$ for the NE classes (and not for the non-NE class) on a phrase-level basis where a phrase (token sequence) is true positive iff each token of the etalon phrase is labeled correctly. Then the results of the classes were aggregated by a sample-size weighted average to get the system-level F-measure.

$$P = TP/(TP + FP),\ R = TP/(TP + FN),\ F_{\beta=1} = \frac{2 * P * R}{P + R},$$

where P,R,TP,FP,FN stand for precision, recall, true positive matches, false positive matches and false negative matches, respectively.

We employed two baseline methods on the Hungarian NER dataset. The first one was based on the following decision rule: *For each term that is part of an entity, assign the organization class*. This simple method achieved a precision score of 71.9% and a recall score 69.8% on the evaluation sets (F-measure of 70.8%). These good results are due to the fact that information about what an NE is (and is not) and the characteristics of the domain (in business news articles the *organization* class dominates the other three) were added to the baseline algorithm. The second baseline algorithm selected the complete unambiguous named entities appearing in the training data and attained an F-measure score of 73.51%. These results are slightly better than those published for different languages, which is due to the unique characteristics of business news texts where the distribution of entities is biased towards the organization class.

Table 3.1 contains the results achieved by the three learners [1]. The results for the ANN and C4.5 are quite similar to each other. The recall for the SVM were significantly lower on three classes outside *organisation* (where it is significantly greater) compared to those for ANN and C4.5. This means that it separated the NE and non-NE classes well but could not separate the NE classes themselves; it predicated too much *organisation*.

[1] Earlier evaluations like MUC [14] used the token-level F-measure, which is a less strict one.

The preference of the majority NE class might be due to the independence of the SVM to the distribution of the sample points.

	Precision / Recall / F-measure (%)		
	ANN	C4.5	SVM
Location	80.9/67.9/73.8	79.8/69.9/74.5	90.2/30.2/45.2
Organization	88.1/89.9/89.0	87.8/89.2/88.5	87.5/94.8/ 91.0
Person names	81.8/80.4/81.1	77.2/77.2/77.2	80.3/70.7/ 75.2
Miscellaneous	81.3/60.2/69.2	78.8/60.7/68.6	92.1/56.7/ 70.2
OVERALL	**86.1/83.9/85.0**	**84.7/83.7/84.2**	**84.1/83.9/ 84.0**
IMPROVEMENT TO THE BEST BASELINE	**8.3/24.0/11.5**	**6.9/23.8/10.7**	**6.3/24.0/ 10.5**

Table 3.1: Results of various learning models on the Hungarian NER task.

Attribute selection was employed (the statistical chi-squared test) to rank the features we had, in order to examine the behaviour of the learners in less noisy environments. Our intuition was that among the features there were several which were not implicative, i.e. they just confused the systems. After performing ranking we examined the performance as a function of the number of features used by using the C4.5 decision tree learner for classification (wrapper feature selection approach). We found that keeping just the first 60 features increased the overall accuracy of the tree because we got rid of many of the features that had little statistical relevance to the target class. C4.5 in the filtered feature space achieved an 85.18% F-measure, which was better than any of the three individual models using the full feature set. ANN and SVM on the other hand produced poorer results on this reduced feature set (with F-measure scores of 83.87% and 83.23% respectively) and it shows that these numeric learners managed to capture some information from these less significant features. We suppose that the tree performs worse in the presence of the less indicative features because in the case of small object sets it chooses them for cutting (i.e. it overfits). We think that this effect could be minimised by fine-tuning the decision tree pruning parameters.

In line with the above-mentioned experiments we chose to employ C4.5 in our further experiments for the following reasons:

- The results obtained were comparable to other learners, but it had a significantly lower training time.
- Its output (the decision tree) is human-readable so it is readily interpretable and can be extended by domain experts.
- We expect that the optimal decision rule set of most IE tasks are similar to AND/OR rules built on the mainly discrete feature set, not hyperplanes or activation functions.

- The general criticism against using decision trees [49] is that the splitting method favours features with many values. In the IE tasks the features usually have at most 3-4 values, hence in our applications this bias is naturally avoided.

3.1.6 Decision tree versus Logistic Regression

The *Logistic Regression* classifier [50] (sometimes called the *Maximum Entropy Classifier*) also has the characteristic – which is favourable for IE tasks – of handling discrete features in a suitable form (many existing implementations[2] work exclusively on binary features).

Generative models like the *Naive Bayes* [51] are based on the joint distribution $p(\mathbf{x}, y)$, which requires the modeling of $p(\mathbf{x})$. In real-world applications $\mathbf{x}$ usually consists of many dependent and overlapping features, hence its proper modeling is intractable (Naive Bayes models $p(\mathbf{x})$ using the naive assumption that the features are independent of each other). Logistic Regression is a discriminative learning model which directly models the conditional probability $p(y|x)$, thus avoids the problem of having to model $p(\mathbf{x})$ [52]. Its basic assumption is that the conditional probability of a certain class fits a logistic curve:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{\sum_{j=1}^{|F|} w_{y,j} x_j\},$$

where $w_{y,j}$ are the target variables and $Z(\mathbf{x}) = \sum_y \exp(\sum_{j=1}^{|F|} w_{y,j} x_j)$ is a normalisation factor.

We experimentally compared Logistic Regression and C4.5 using the metonymy resolution dataset (see Section 2.3.5). A rich feature set was constructed for this task [3] which included

grammatical annotations: the grammatical relations (relation type and headword) given in the corpus was used and the set of headwords was generalised using lexical resources,

determiner: the type of nearest determiner,

number: whether the target NE is in plural or singular form,

word form: the word form of the target NE.

Our resulting system which made use of Logistic Regression achieved an overall accuracy of 72.80% for organisation name metonymies and 84.36% for location names

[2] e.g. http://maxent.sourceforge.net and http://mallet.cs.umass.edu

on the unseen test set. Our team[3] had the highest accuracy scores for the metonymy resolution shared task at SemEval-2007 in all six subtasks [28]. We should add that even our results just slightly outperformed the baselines, which implies that the issue of resolving metonymies is rather complex and as yet unsolved.

We compared Logistic Regression[4] and C4.5 with a moderate parameter tuning using a five-fold test validation on the training set as we only had access to this dataset during the shared task (another short experimental comparison will be described in Section 6.3.4). We found that Logistic Regression outperformed C4.5 by 2-3% in every subtask. We suppose that this might be due to the quite different nature of the feature sets of the NER and metonymy datasets. In the standard NER problem we have several hundreds of mainly binary features where the decision tree can choose the most important ones (which form a small subset of F). On the other hand, the feature set of the metonymy resolution task has a very complex inter-dependence among features.

Logistic Regression has one more advantage. It estimates $p(y|\mathbf{x})$, so the distribution on the class labels is given. These distributions can be very important for a ranking application and can be used to perform a precision-recall tradeoff by applying thresholds on them. Decision trees can provide class probabilities as well by calculating them from the homogeneity of the object on the appropriate leaf, but the results are less precise. Hence, for discrete features we suggest using Logistic Regression when the number of features is moderate and we expect that there will be a complex dependence among them, or posterior probabilities are required. In other cases a decision tree can be employed because it achieves similar results but is significantly faster and its output is easily interpretable.

3.2 Sequential models

In sequence labeling, the output of the system is a prediction for the whole sequence of instances so we should not compute a prediction for each instance separately. In IE tasks, this approach labels a sentence, i.e. its output for one prediction is a sequence of labels for the tokens of the sentence. It does nit assume that each element of the sequence is independent. For example the prediction on the second token of an NE phrase cannot be independent of the prediction on the first token. To handle inter-token dependencies a more complex model is required, so sequential models have a worse training time complexity compared to token-based ones.

Token-based IE models are capable of taking into account the relationships between consecutive words as well, collecting the relevant features into a window of appropriate

[3] A joint team of the Media Research Center at the Department of Sociology and Communications of BUTE and University of Szeged, Department of Informatics.

[4] We used the implementation http://homepages.inf.ed.ac.uk/s0450736/maxenttoolkit.html.

size. This means that the dependence is incorporated in the feature set (e.g. it can contain a feature like *the previous token is listed in a gazetteer*). This type of modelling provides the opportunity of sampling from individual tokens. This is necessary for creating a balanced training data, which is sometimes beneficial to learning algorithms.

The problem arises when we try to employ the class labels of the neighbouring tokens. We used an online evaluation in the NER experiments: after predicting the classlabel of the tth token of the sentence we modify the feature values of the $(t+i)$th tokens ($0 < i <= s$, where s is the window size). We can directly utilise previous class labels by this approach. This can be generalised to consecutive labels as well by performing more iterations of prediction on the sentence. The main problem with online evaluation is that during the training phase it utilises the previous labels from the gold standard annotation, but during the test phase the previous labels could be wrong, which leads to the infamous *"label bias"* problem [53].

Conditional Random Fields (CRF)[53] is the state-of-the sequence labeling learner. It models conditional probabilities in the form of $p(\mathbf{y}|\mathbf{x})$ instead of local estimates $p(y_t|\mathbf{x}_t)$ (where the t here refers to sentence positions):

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{\sum_t \sum_{j=1}^{|F|} w_j f_j(x, y_t, y_{t-1})\}$$

Only the simple chain structured random fields are tractable as training is done by running the forward-backward algorithm through several iterations. The training (finding the optimal $\mathbf{w}$) is usually performed by maximising the conditional log likelihood

$$\max_w \ell(w) = \sum^N \log p(\mathbf{y}|\mathbf{x})$$

We showed experimentally on the NER tasks [54] that our online evaluation-based approach is competitive with the sequence labeling algorithms. The CRF achieved a better accuracies than C4.5 on Hungarian, English and German NER using the same feature set, but the absolute differences in $F_{\beta=1}$ were below 2% [54]. On the other hand, we emphasise once again the advantages of using decision trees, namely the speed of training and easy interpretability of the output results.

3.3 Related work

While just one learning system [55] and seven rule-based classifiers were submitted to the last MUC contest [14] (in 1997), only statistical algorithms were entered in the shared task of CoNLL-2003, where the most frequently applied techniques [22] were Logistic Regression [45] and Hidden Markov Models (HMM) [56]. On the other hand,

in the Natural Language Processing in Biomedicine and its Applications (JNLPBA) bio-entity recognition task [48], the Support Vector Machine [57] – used in isolation or in combination with other models – was the most popular one.

Outside the shared tasks several token-level NER systems were investigated like Decision Trees [58], Logistic Regression [59] and Support Vector Machines [60]. The first results with sequential models [61] were obtained by applying HMM [62] in the nineties. A few years later the Maximum Entropy Markov Model (MEMM) [63] was presented, which is a sequential extension of Logistic Regression and maximizes the likelihood of being in a state (NE class) conditioned on the previous token's label and the current token. Later, NER was one of the first fields of application for CRF [64] as well.

Previous research on Hungarian NER included expert rule-based approaches mainly because no labelled corpora was available for training statistical models. To the best of the author's knowledge, the first NE tagger was developed by researchers of the Hungarian Academy of Sciences at the Institute for Linguistics [65]. The system exploits regular patterns that capture Named Entity phrases in Hungarian texts. Later on the NER module of the HumorEsk [66] expert-rule-based Hungarian syntactic parser was built based on Gábor et al.'s system.

The only other statistical NER system that was trained and tested on the same Hungarian corpus using similar evaluation metrics is the system created by the Technical University of Budapest [47]. They trained the Logistic Regression classifier based on their own feature representation and achieved a slightly higher F-measure than that for our model. This difference can be attributed to the different features used and the learning method used by them.

3.4 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 NE reference corpus [2], which allowed researchers to investigate statistical approaches. Together with his colleagues, the author constructed a Machine Learning-based NER system [1] and a metonymy resolution system (GYDER) [3]. Our classification systems achieved results on international reference datasets which are competitive with other state-of-the-art NER taggers while they have several distinct characteristics. The construction of the SzegedNE corpus is an inseparable contribution of the authors of [2]. The major contribution of the author here is an experimental comparison of the token-based versus sequential approaches, and several classification algorithms. He made several suggestions about which algorithm should be used in a given learning environment as well.

For any Machine Learning algorithm there is at least one task where it achieves

good accuracy and another task where it does not. This phenomena known as the *No Free Lunch Theorem* [67]. Hence different methods should be directly compared on just one particular dataset. Even so, the author argues that IE tasks share the same common characteristics and NER (along with metonymy resolution) can be regarded as a prototype task, so the conclusions drawn from it can be generalised to other tasks. When turning to the issue of making a comparison several points arise:

- The absolute difference between the algorithms' accuracy was moderate (below 2%) using the same data representation.
- The decision tree has the most favourable time complexity.
- Only the output of the decision tree is directly interpretable by users.
- The generative probabilistic models – like Logistic Regression and CRF – output a probability distribution on the class labels which could be a confidence measure for an application.

Overall, we recommend using decision trees in the development phase (experiments) of an application because of its training time, ease of interpretability and use of generative models (Logistic Regression in classification and CRF in sequence labeling tasks) in the final versions.

Chapter 4

Combinations of learning models

When attempting to solve classification problems effectively it is worth applying various types of classification methods as the *No Free Lunch Theorem* [67] states that there is no single learning algorithm which in any given task always achieves the best results. The methods can be employed separately and in combination. The success of hybrid combination approaches lies in tackling the problem from several aspects, so algorithms with inherently different theoretical bases are good subjects for voting and for other combination schemes.

There are several well-known combination (also know as *meta-learning*) algorithms in the literature that can lead to a 'better' model – in terms of classification accuracy – than those serving as a basis for it, or can significantly decrease the CPU time of the learning phase without loss of accuracy. Decreasing the learning time by dividing the training data among different learners (known as *bagging* [68]) would be necessary for a corpus with a size of millions of tokens. Since in our case the time needed for learning was not really relevant (we showed in the previous chapter that fast methods like C4.5 can achieve competitive results), we concentrated on improving the accuracy of the system.

We examined several extensions of our NER model on English and Hungarian datasets. These extensions are combinations of learning models (we call these models *base-learners* here) in various ways and are described in this chapter.

4.1 Stacking

We experimented on the utilisation possibilities of the combination of base-learners described in Section 3.1 on the Hungarian NER dataset. One of the simplest ways of combining multiple classifiers is *Stacking* [69]. In Stacking, the final output is got by an arbitrary g combiner function based on the base-learners' predictions: $y = g(d_1, d_2, ..., d_n)$. The theoretical assumption underlying Stacking is that hypotheses made by inherently

different learning methods usually cover different parts of the solution space well, which means that their error surface is not necessarily the same. This way hybrid methods can combine the good characteristics of the individual models, leading to a better overall performance. The combiner function can be a manually constructed decision rule or a machine-learnt decision function (in this case a development dataset is required). Note that it can be regarded as a generalisation of voting, where the combiner function is a linear combination of the base predictions.

To combine the results of the three base-learners (see Section 3.1.5) on the Hungarian NER dataset we used the Stacking method. The decision function we applied to integrate the learnt hypotheses of the C4.5, ANN and SVM was the following: *if any two of three learners' output coincide we accept it as a joint prediction, and use the forecast of the best performing model (the ANN) otherwise (if three different answers are given by the three models)*. The results of the Stacking method compared to the best base-learners is shown in Table 4.1.

	ANN	Stacking
Location	75.8	77.7
Organization	94.5	94.9
Person names	83.4	83.9
Miscellaneous	69.8	70.1
OVERALL	**85.0**	**87.53**

Table 4.1: The results of the Stacking method on the Hungarian NER task.

This Stacking scheme performed better than any base-learner and had an F-measure of 87.53%. This is a significant improvement (2.53%) compared to the best base-learner (ANN), with a 16.9% decrease in the error rate relative to the neural network. The experimental results confirm that even the base-learners with lower accuracy have added value in a combination scheme. We used the same Stacking approach to combine our English NER system with other outer systems, whose results are described in Section 4.5.

4.2 Boosting

Boosting [70] was introduced by Shapire as a way of improving the performance of a base-learner algorithm. In Boosting a sequence of base-learners are trained on the mistakes of the previous learners. The most widely used Boosting strategy is AdaBoost [71], which generates a set of base-learners by re-weighting instances on the basis of rightly and wrongly classified samples on the original training dataset. The final decision is made using a weighted voting schema for each classifier, where the weights

are calculated based on the accuracy of the base-learners. This scheme is many times more accurate than the original base model.

We examined AdaBoost on the Hungarian and English newswire datasets and the English medical de-identification (Section 2.3.4) NER datasets. 30 iterations of AdaBoost were performed for each experiment because further iterations on the CoNLL-2003 development set brought only slight improvements in the F-measure (less than 0.05%). We used a decision tree as the base-learner because it is fast (the number of Boosting iterations multiplies the training time) and it is the most sensible for the re-weigthing of the training sample among the three base-learners SVM, ANN and C4.5.

	Hungarian newswire	English newswire	English medical
C4.5	84.04	79.24	95.05
AdaBoost+C4.5	92.77	84.81	96.60
error reduction	54.6%	26.8%	31.3%

Table 4.2: The added value of Boosting.

Table 4.2 [4][5] summarises the results achieved by AdaBoost, whose error reduction is striking. Figure 4.1 shows the change in F-measure achieved on the Hungarian NER dataset – evaluated on the training and test set – as a function of AdaBoost iterations. We note that it achieves significantly better results, even after the first iteration, than the best base-learner (the ANN with 85%). The results also experimentally confirm the theoretical property of Boosting that its improvement in accuracy has a logarithmic trend [2].

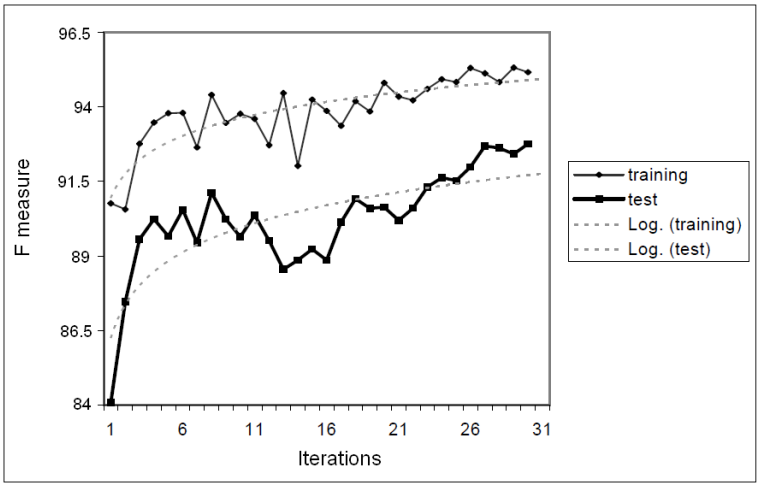

Figure 4.1: The accuracy of NER models during the AdaBoost iterations.

4.3 Feature set split and recombination

We built a rich feature set for the NER tasks which were introduced in Section 3.1.4. We exploited it in the following way. First, several overlapping smaller feature sets were selected from the whole set. These sets describe the tokens from different perspectives and an accurate model can still be built on them. Then learning models on each smaller feature set were employed and their predictions in a Stacking approach were used to make the final decision. This procedure can be regarded as a bagging method where instead of train models on different entity sets the bags contain the same dataset, but from a different view as different features describe it.

In the NER experiments six semantic groups of features were formed:

- orthographical features,
- gazetteers of unambiguous NEs from the train data,
- dictionaries,
- frequency information,
- phrasal information,
- contextual information.

Using these six groups we trained a decision tree for each possible subset of the groups (63 base-learners) and evaluated them on the English development dataset. Naturally not every subset described the NER problem equally well. We used C4.5 trees here because their training is very fast and we assumed that the differences between single trees would not change significantly while boosting them. We evaluated these models on the CoNLL development set.

We decided to keep the 5 best of the 63 base-learners for CPU time consumption reasons. We trained a Boosted decision tree using each of these five (overlapping) feature sets and then recombined the resulting models in a Stacking scheme: *if any three of the five learners' outputs coincided we accepted it as a joint prediction, with a forecasted 'O' label referring to a non-Named Entity class otherwise*. This cautious stacking scheme is beneficial to system performance as a high rate of disagreement often means a poor prediction rate. For a phrase-level evaluation it is better to make the kind of mistakes that classify an NE as a non-Named Entity than place an NE in a wrong entity class (the latter detrimentally affects precision and recall, while the former only affects the recall of the system).

As the same type of features were used in English and Hungarian experiments, we decided to investigate the same five subsets of features on the Hungarian language as well (i.e. we did not train 63 base-learners for Hungarian, but used the five which

performed best on the English data). The results achieved by this *Feature split and recombination* method are shown in Table 4.3. The baseline algorithm here selects complete unambiguous named entities appearing in the training data.

	English	Hungarian
Baseline	59.61	70.99
Full feature set	84.81	92.77
The five models	81.3-84.6	88.1-93.7
Recombination	86.90	94.69

Table 4.3: The results of Feature split and recombination.

The feature set split and recombination method on both languages yielded scores about 2% better than a single model trained on the whole feature set.

4.4 Post processing

The output of a statistical model usually contains errors which can be recognised (and avoided) by humans. We formulated several "trivial error correcting" rules based on the typical errors of the system observed in the development set. These rules were applied on the forecasts of the system, thus can be regarded as a cascade combination of the statistical and rule-based systems. These simple post-processing rules can bring about some improvements in the overall system accuracy.

Take, for instance, full person names which consist of first names and family names, which are easier to recognize than 'standalone' family names that refer to a person (e.g. "John Nash" or "Nash"). Here if we recognize a full name and encounter the family name later in the document we simply overwrite its label with a person name. This is a reasonable assumption that holds true in most cases. Certain types of NEs rarely follow each other without any punctuation marks, so if our term-level classification model produces such an output we overwrite all class labels of this sequence with the label assigned to its head. Acronym words are often easier to disambiguate in their longer phrase form, so if we find both in the same document we change the prediction given for the acronym when it does not coincide with the encountered longer form.

	Family names	Rare sequence filter	Acronym
English	+0.25	+0.16	+0.47
Hungarian	+0.07	-0.21	+0.22

Table 4.4: Improvements of the post processing steps on the test datasets based on the previous step.

These three post processing rules do not involve any learning or adaptation, but were simply evaluated on the development dataset and found to be useful for both English and Hungarian – although their improvement on the Hungarian NER system was only marginal and the rare sequence filter was not significant (see Table 4.4). Similar and other simple post-processing steps have been incorporated in several NER systems (for example in [72], which came second in the CoNLL-2003 contest).

4.5 The complex NER system

The approaches described in the previous sections are the building blocks of our complex NER system sketched in Figure 4.2. Firstly, the full feature set – which contains mostly nominal attributes – is extracted from the raw text and split into the five (overlapping) chosen subsets. AdaBoosted C4.5 decision trees are trained on these feature subsets and the Stacking combination of their forecasts is the output of our "individual" system. We investigated the combination of this system with those of the best two NER systems [73][72] of the CoNLL-2003 contest ("external" systems), applying the Stacking rule *if any two of three learners' output coincide we accept it as a joint prediction, and use the forecast of the best performing model [73]; otherwise (if three different answers are given by the three models)*. Lastly, this forecast is post-processed and the text is labeled.

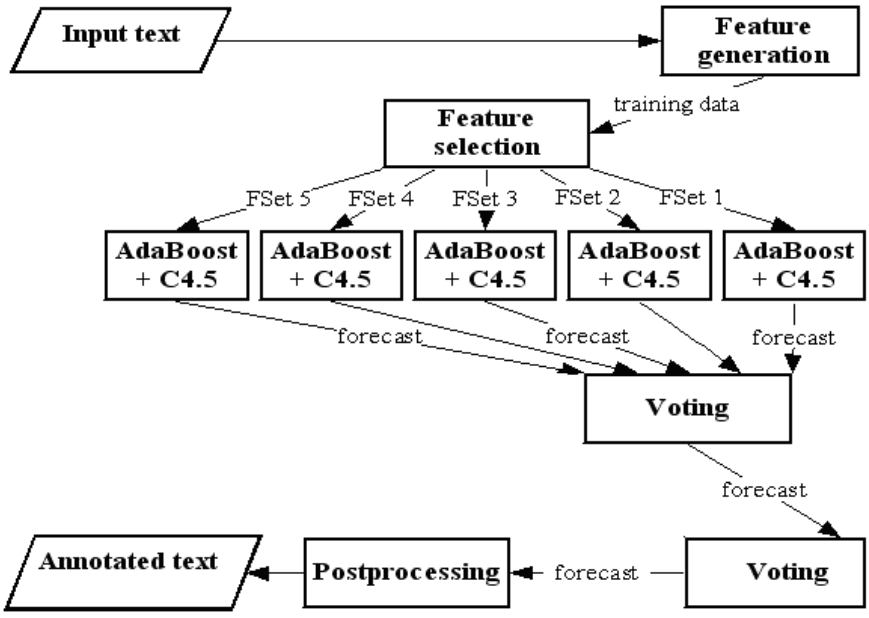

Figure 4.2: Outline of the structure of our complex NER model.

The best participating system of the shared task [73] achieved an 88.76% phrase-level F-value on the evaluation set. Our individual system achieved 89.02%, which corresponds to a 2.32% error reduction relative to the best model known that was

tested on the same data (although this difference is not statistically significant). We should point out here that the system in [73] made use of the output of two externally trained NE taggers and thus the best standalone model in the shared task was that described in [72]. When compared to it, it had an error reduction of 6.08%.

Our system combined with the two external systems performed significantly better (having an F-measure score of 91.41%) than the best hybrid NER system reported in the shared task paper which employed the 5 best participating models (F-measures of 90.3%). This meant a significant (11.44%) reduction in misclassified NEs. The successful applicability of our model in such a stacking approach is presumably due to ours having an inherently different theoretical background, which is usually beneficial to combination schemes. Our system uses Boosting and C4.5 decision tree learning, while the other two systems incorporate the Robust Linear Classifier, Transformation-Based Learning, the Maximum Entropy Classifier and the Hidden Markov Model.

4.6 Combination in a bootstrapping environment

We investigated co-training, a learning model combination technique in a bootstrapping environment. One of the most common approaches in semi-supervised learning (see Section 2.2.3) is bootstrapping. In bootstrapping, the unlabeled data is used as a pool where new training examples are chosen from [74]. This training set expansion is performed by a classifier – in several iterations – which first labels the unlabeled set and chooses the most reliable entities (called as *self-training*). In *co-training* two or more different classifiers are used for predicting unlabeled data, then they "teach" each other via the most reliable instances from the unlabeled pool [74].

We experimentally compared self-training and co-training on the Hungarian and English NER datasets. In the English task we used the CoNLL-2003 corpus but we evaluated our methods on the development set of the contest, because the evaluation set differed in its characteristics from the train set. One of the aims of this contest was to discover the usefulness of unlabeled texts, but none of the participating systems made use of them in a sophisticated way [22]. The database contained over 18 million unlabeled tokens which were used in our experiments. In the Hungarian NER corpus we do not have unlabeled texts from the same source as this corpus. Our investigations on other raw texts (from the economy domain) were unsuccessful. Hence we followed the transductive approach (see Section 2.2.3) in Hungarian semi-supervised experiments; that is, the evaluation dataset was used as unlabeled text.

We employed the CRF and the boosted C4.5 learning algorithms as the diversity of these two models makes them good candidates for combination. Figure 4.3 shows the results obtained by self-training (dotted line) and co-training (continuous line) with different sizes of labeled training data. The baselines (the zero point on the X axis) were

the accuracies of the supervised CRF model. The accuracy score of co-training could be improved if we did not use every predicted sentences but defined a confidence threshold of the decision tree for choosing reliable sentences. Obviously the lower the threshold, the lower the ratio of sentences which match the criteria, and thus a larger unlabeled initial database would be required. Figure 4.4 shows two confidence level settings: the continuous line (the same as in Figure 4.3) represents a confidence threshold of 10^{-3}, while the dotted line represents a threshold of 10^{-10}.

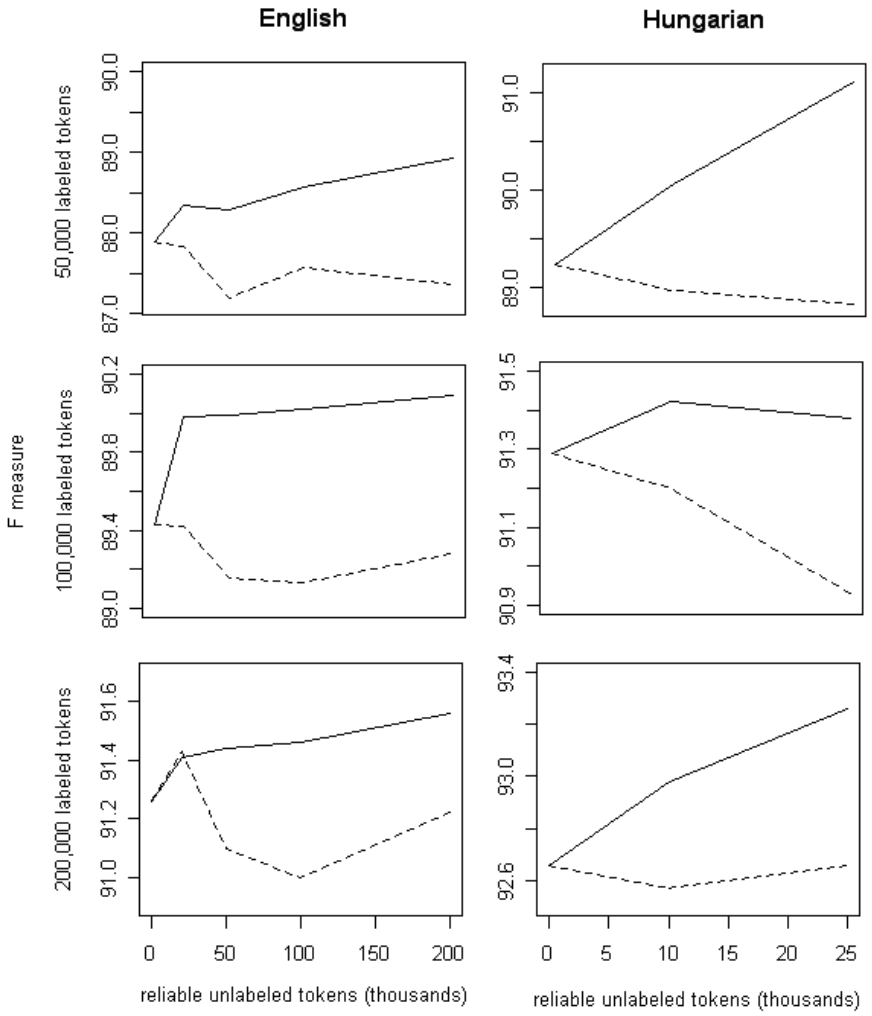

Figure 4.3: Self-training (dotted) and co-training (continuous) results on the NER tasks.

The key conclusion of these experiments is that the increasing trend remains stable even when we use large unlabeled datasets as well and we could achieve slightly better results by co-training with 100,000 labeled tokens (with an F-measure of 91.28%) instead of using the supervised model with 200,000 labeled tokens (91.26%). In order to get such an accuracy score with 100,000 labeled tokens we gathered 23,507 reliable

sentences (400,000 tokens) from a raw text of 3 million tokens.

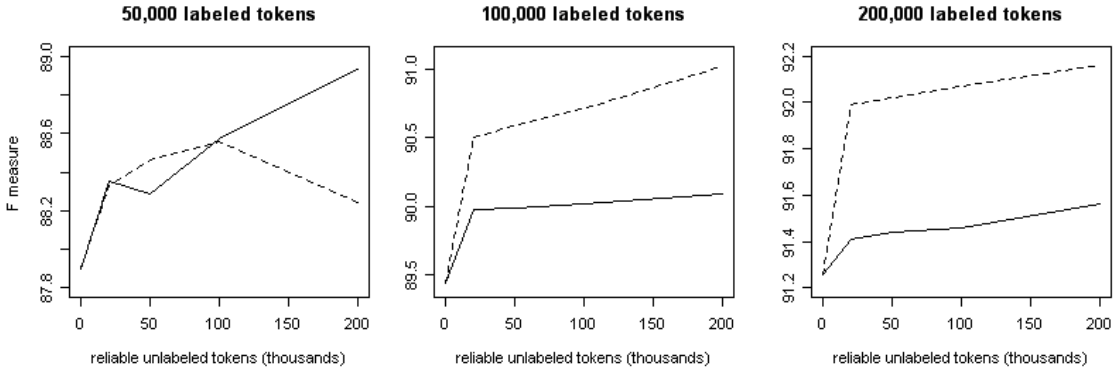

Figure 4.4: Co-training results with confidence thresholds of 10^{-3} (continuous line) and 10^{-10} (dotted).

Figures 4.3 and 4.4 show that, by using unlabeled texts, the results of a supervised model can be improved significantly with every size of labeled training data both in English and Hungarian NER tasks. Along with these nice results we must mention some poor ones as well. Co-training was not robust when we evaluated it on the evaluation set of the CoNLL-2003 contest (instead of the development set). In the case of 100,000 labeled examples plus 200,000 raw "reliable" tokens, the supervised model achieved an F-measure of 83.58%, while co-training with a 10^{-3} confidence threshold gave just 83.21%. The model with a 10^{-10} confidence level improved the accuracy but not by a significant amount (F-measure of 83.62%). Similar results were achieved when we investigated bootstrapping methods with raw Hungarian economy texts obtained from a source different from that of our corpus. We found that neither self-training nor co-training could achieve better results than the supervised CRF. Based on these experiments we came to the conclusion that the training set, the evaluation set and the unlabeled dataset as well should have similar characteristics in a well functioning bootstrapping system or automated domain adaptation has to be performed.

4.7 Related Work

Various model combination techniques have been used since the early '90s (for an excellent overview see, Chapter 15 of [49]). A system that made use of AdaBoost and fixed depth decision trees [75] came first on the CoNLL-2002 conference shared task for Dutch and Spanish, but gave somewhat worse results for English and German in 2003 (it was ranked fifth). We have not found any other competitive results published so far for NER using decision tree classifiers and Boosting.

We are aware of two published works which present procedures similar to our feature set split and recombination technique. Lazarevic [76] chose the feature sets randomly then the final decision is a linear combination of the models trained on these feature sets (they solved an unsupervised outlier detection problem). Sutton et al. [77] dealt with the NER task focusing on the weight undertraining problem. Here, the feature set is divided into two disjunct sets (character n-gramm based ones and lexicons) then a CRF-specific combination is applied. Our feature set split and recombination technique is more general and the exploitation of a fast learning algorithm provides a more efficient approach compared to random selection and fixed bags.

As far as we know our results are the first real semi-supervised ones on the CoNLL database; H. Ji and R. Grishman [78] provided self-training results with HMM on the ACE04 NER task. Co-training has been applied with success in several IE tasks, for example in [79] and [80].

4.8 Summary of thesis results

In this chapter, the author introduced experiments on combination schemes of different Machine Learning algorithms. The building and testing of less frequently applied algorithms is always worth doing, since they can have a positive effect when combined with popular models. The author empirically verified this statement in the Hungarian and English NER tasks, applying Stacking [1][4], Boosting [4] and co-training [54]. He introduced a novel combination approach, called *Feature set split and recombination* [4], which exploits the rich feature set.

These combination schemes play a key role in the construction of the complex NER system by the author with his colleagues. This NER system is competitive with the published state-of-the-art systems (an F-measure of 89.2% and 94.77% for English and for Hungarian, respectively). It has a different theoretical background compared to the widely used sequential ones, which makes it an excellent candidate for a combination scheme with external NER systems. The main result of this chapter can summarised as follows. Simple combination schemes are worth employing because they usually bring about a significant accuracy improvement. Moreover, the scores are usually better than those achieved by employing more sophisticated but time-consuming stand-alone learning algorithms.

Chapter 5

On the adaptability of IE systems

In the previous two chapters we introduced a complex supervised NER system which achieved good results on several particular datasets. Here, we examine the adaptability of such systems, i.e. how the performance of these systems changes on new datasets which have a different language or in domain. We will also describe our experiments in adaptation-related investigations on several different IE tasks.

5.1 Language (in)dependence

The NER task was introduced during the nineties as a part of the shared tasks in the Message Understanding Conferences (MUC) [14]. The goal of these conferences was the recognition of proper nouns (person, organization, location names), and other phrases denoting dates, time intervals, and measures in texts collected from English and Japanese newspaper articles.

Later, as a part of the Computational Natural Language Learning (CoNLL) conferences [81], a shared task dealt with the development of NER systems for multiple languages (first introduced in [82]), which were able to correctly identify a person, an organization and location names, along with other proper nouns treated as miscellaneous entities. The collection of texts consisted of newswire articles, in Spanish + Dutch and English + German, respectively.

Other major languages like Arabic [83], Chinese [84], French [85] or Russian [86] have been studied in the literature – outside of the shared tasks – as well. More recently, less frequently spoken languages like Bulgarian [87], Basque [88], Polish [89] and Hungarian [1] have also gained attention.

In our English and Hungarian experiments, essentially the same features were utilised [4]. We used the same type but language-dependent dictionaries (first names, company types, locations etc.) and re-collected the gazetteers of unambiguous NEs from the train data. We applied a different categorization of the linguistic information (POS and

chunk codes) as well. The format of the English articles enabled us to introduce two new features, namely *topic code* and *document zone* features (like title, body). All other features, learning methods and combination schemes were the same as before.

Tables 4.2, 4.3 and 4.4 show that – with these minor modifications – the NER system has the same characteristics, its components have similar added values and can achieve good accuracies (an F-measure of 89.2% and 94.77% for English and for Hungarian, respectively) on both languages. These results are quite satisfactory if we take into account the fact that the results for English are by far the best known, while NLP tasks in Hungarian are usually more difficult to handle because Hungarian has many special (and from a statistical learning point of view, undesirable) characteristics. Our NER system remains portable across languages as long as language specific resources are available; and it can be applied successfully to languages with very different characteristics.

5.2 Medical NER

As medical de-identification (see Section 2.3.4) is essentially a NER task we participated in the I2B2 shared task with our NER model adapted to medical records [5]. Now we will present this adaptation process.

5.2.1 Adaptation of the NER system

We adapted our newswire NER model to the medical de-identification task in three steps. First, domain-specific features were added. We extended the feature set used for the newswire domain by introducing two new features: (i) regular expressions that try to cover the well-formulated classes like *phone number*s, which did not occur in the CoNLL task and (ii) our model can infer knowledge from the structure of the document using the common headings observed in typical discharge records (the most frequent subject headings were extracted from the training set).

An important characteristic of the task (domain) was that the NEs were artificially re-identified. Thus the contextual information had to play a key role here, unlike that in the newswire domain. The use of trigger words – which are the most important contextual features – is not straightforward, however, 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 the subsequent tokens of *tagged phrases* and used a wider window for this feature (*phrase trigger*); and third we collected the uni-, bi- and trigramms around the phrases of the train texts (*trigramm trigger*).

The collected lists for each of the three cases were filtered according to their frequency and information gain on the class labels. A significant difference in the predic-

tions was noticed in experiments where only the use of triggers was changed, hence we decided to combine their forecasts to exploit all their advantages. The Stacking function we used to integrate the three hypotheses (learnt by varying our use of triggers) was again the following: *if any two of the three learners' outputs coincided we accepted it as a joint prediction, and forecasted a 'O' label for a non-PHI entity class otherwise*. Thus this kind of feature set combination can be regarded as an alternative for the Feature set split and recombination procedure introduced in Section 4.3 (this is the second domain-adaption step). Here we applied the AdaBoosted C4.5 algorithm again as we did for the newswire domain (Section 4.2).

We exploited another characteristic of the task as well, namely the semi-structured form of the medical records. The structured parts of the text can be processed more easily than the flow text and the named entities in the record fields can occur in other parts of the text in the same or similar form. To utilise this fact we tagged only trusted NEs (appearing in document sections belonging to certain headings) in the first training phase. We considered a heading unambiguous if its cross-class Shannon entropy was less then 0.1 on the train set. The NEs found in this first phase and their acronym became trusted phrases and their lists were added to the feature set of the second training phase. Fortunately, the structured parts of data usually contained fully formed phrases and thus incorporating PHI found there proved to be beneficial to the model.

In the last phase of the system a post-processing step were performed: we standardised the tagging of the same phrases because our token-based classification approach can fail when tagging whole phrases. We collected all the predicted phrases and overwrote every occurrence of them with the predicted class of the longest matching phrase.

5.2.2 Experimental results

We divided the train data into ten parts (it was always cut on the document boundaries), and carried out a ten-fold cross validation on these subsets. We used the AdaBoosted C4.5 as a baseline method here but with a highly reduced feature set. We kept only those features which were thought to be the most significant (namely triggers, initial letter, predicted class of previous token and other four features). Note that the baseline algorithm which selects complete unambiguous named entities appearing in the training data could not be applied here because the corpora were artificially re-identificated.

Table 5.1 lists the accuracies of the base learners and baseline methods. Each value in this table is the size-weighted average of the ten train-test folds. All of the three models learnt on the different trigger features significantly outperformed the baseline method, which demonstrates the real value of our enriched feature set.

The accuracies of the three trigger methods are somewhat similar to each other but their predictions are far from identical; consequently they generally perform well

	P	R	$F_{\beta=1}$
Baseline	81.03	79.42	80.15
Token trigger	97.48	95.74	96.60
Phrase trigger	97.17	95.78	96.47
Trigramm trigger	97.56	95.89	96.72
Stacking	98.02	95.82	96.91

Table 5.1: Result of the trigger methods.

where the other two fail. The accuracy increased in their Stacked combination, which confirms this point.

	P	R	$F_{\beta=1}$
Staked model	98.0	95.8	96.9
Trusted NEs	97.8	96.4	97.1
Post-process	98.1	96.7	97.4

Table 5.2: The added value of the two post-processing steps.

Table 5.2 shows the results we got after performing the two domain-specific post-processing steps described in the previous subsection. Owing to the trusted phrases, the second trained model tagged more instances as PHI than the first model (resulting in a higher recall), but several mistakes were made among these tagged phrases (the precision decreased). Because the phrase-level evaluation metric penalised the partially tagged phrases twice (it affects both precision and recall), the standardisation of the predicted phrases increased both the precision and recall.

	P	R	$F_{\beta=1}$	#pred	#etal
ID	99.5	99.2	99.33	3670	3678
AGE	100.0	91.7	95.00	12	13
DATE	99.3	99.3	99.25	5191	5193
PHONE	100.0	97.3	98.61	170	175
DOCTOR	96.9	94.9	95.88	2635	2690
PATIENT	96.6	95.9	96.21	683	685
HOSPITAL	95.5	90.1	92.69	1634	1736
LOCATION	75.8	57.1	63.79	108	144
all	98.1	96.7	97.41	14104	14314

Table 5.3: The per class accuracies and frequencies of our final, best model.

Table 5.3 provides an overview of the final accuracies obtained from each PHI class separately. The most accurate ones are the well-formed classes (*id*, *age*, *date*, *phone*), with an $F_{\beta=1}$ measure above 98.5%. This is mainly due to the fact that they can be processed by simple regular expressions and they occur in the same form in the unstructured texts, as noticed in the fields of the records.

We made bad predictions on the class *location*, but considering the complexity of its recognisability and the amount of available training examples (we had fewer than 200 examples available for this class) it really seemed to be an intractable problem. The performance of the classes *doctor*, *patient*, *hospital* were similar to those we published previously – and described in the related NER works – for Named Entity classes on newswire articles. The better results achieved here on the de-identification task were probably due to the semi-structured nature of the documents.

Every learnt model in our experiments was significantly better on precision than recall. This may be because they learn just the more certain patterns (this was exploited by our combination schema as well).

We consider the above results fairly promising, as they are probably quite near the inconsistency level of the labelling of data we used. We have no information on the agreement rate of the annotators though, which could explain the precision of training data and give a theoretical upper bound for the accuracy of classification.

Our model achieved an $F_{\beta=1}$ score of 97.4% by ten-fold validation, which demonstrates the success of our adaptation. We would like to emphasize here again that we got this competitive result without any deep parsing information (not even POS codes) and without any domain specific resources. Our success is probably due to the very rich token-level feature set we collected and the combination techniques employed, hence we think that our system could be used (or easily adapted) to other domains as well.

As the system we constructed was trained and tested on a dataset that contained re-identified PHIs, it is quite clear that our model would perform even better in a real-life application. This view is based on two facts. First, several features that would undoubtedly help the recognition of real PHI (like a list of possible first names) fail on the re-identified PHI in this dataset. Second, the artificially increased ambiguity of re-identified PHIs made this task particularly challenging and the results on data like this are probably somewhat poorer than they be in real-life tasks.

5.3 NER on general texts

For the language and domain adaptations described in the previous sections, a manually constructed training set was available. We used our English and Hungarian NER system as a submodule in a sentence alignment tool which was tested on several genre of texts [90], hence we obtained an insight into the performance of our NER systems on general texts where a domain-specific training set was not present.

There are many applications which could benefit from parallel texts (the same content on different languages) like

- automatic translation programs (as Machine Learning algorithms) that are used

as training databases,

- translation support tools that can be obtained from them (translation memories, bilingual dictionaries) and
- Cross-Language Information Extraction methods.

These applications require a high-quality correspondence between text segments like sentences. Sentence alignment establishes relations between sentences of a bilingual parallel corpus. This relation may not have just a one-to-one correspondence between sentences; there could be a many-to-zero alignment (in the case of insertion or deletion), many-to-one alignment (if there is a contraction or an expansion) or even many-to-many alignments.

5.3.1 The role of NEs in sentence alignment

Various methods have been proposed to solve the sentence alignment task. These are all derived from two main classes, namely length-based and lexical methods, but the most successful are combinations of them (hybrid algorithms). *Algorithms using the sentence length* are just based on statistical information given in the parallel text. The common statistical strategies all use the number of characters like Gale & Church's [91] or words like Brown et al.'s method [92] of sentences which models the relationship between sentences to find the best correspondence. These algorithms are not so accurate if sentences are deleted, inserted or there are many-to-one or many-to-many correspondences between sentences. *Lexical-based methods* [93] [94] utilise the fact that if the words in a sentence pair correspond to each other, then the sentences are also probably translations of each other.

A combination of these approaches (hybrid algorithms) utilise various kinds of anchors to enhance the quality of the alignment such as numbers, date expressions, symbols, auxiliary information (like session numbers and the names of speakers in the Hansard corpus[1]) or cognates. Cognates are pairs of tokens of historically related languages with a similar orthography and meaning like parlament/parliament in the case of the English-French language pair. Several methods have also been published to identify cognates. Simard et al. [95] considered words as cognates, i.e. those that had a correspondence with at least four initial letters, so pairs like government-gouvernement should be excluded. These cognate-based methods work well for Indo-European languages, but with languages belonging to different families (like Hungarian-English) or with different character sets the number of cognates found is generally low.

The previously published approaches for a Hungarian-English language pair judged words containing capital letters or digits of equal amount in the text to be the most

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC95T20

trusted anchors, but any mistakenly assigned anchors have to be filtered. Unlike other algorithms, our novel method requires no filtering of anchors because the alignment works with the help of exact anchors like NEs. The following example illustrates the difference between using capitalised words as anchors against using NEs as anchors:

> Az új európai dinamizmus és a változó geopolitikai helyzet arra késztetett három országot, név szerint Olaszországot, Hollandiát és Svédországot, hogy 1995. január 1-jén csatlakozzon az Európai Unióhoz.
>
> The new European dynamism and the continent's changing geopolitics led three more countries - Italy, the Netherlands and Sweden - to join the EU on 1 January 1995.

In the Hungarian sentence there are 5 capitalized words (Olaszországot, Hollandiát, Svédországot, Európai, Unióhoz), unlike its English equivalent which contains 7 (European, Italy, Netherlands and Sweden, EU, January) so using this feature as an anchor would give false results, but an accurate NER module could help it. This example demonstrates as well that cognates cannot be used for a Hungarian-English language pair.

In general, more words are written with a capital letter in English than their Hungarian equivalents. Some examples from the Hungarian-English parallel corpus indeed demonstrate this fact:

- I (én) personal pronoun
- Nationality names: ír söröző = Irish pub
- Location terms: Kossuth Street/Road/Park
- When repeating an expression, the expressions become shorter: pl: European Union = Unió
- Names of countries: Soviet Union = Szovjetunió
- The names of months and days begin with capital letters.

Thus we suggest modifying the base cost of a sentence alignment with the help of NER instead of a bilingual dictionary of anchor words or the number of capital letters in the sentences. This leads to a text-genre independent anchor method that does not require any anchor filtering at all.

5.3.2 The extended sentence alignment algorithm

Our hybrid algorithm [90] for sentence alignment is based on sentence length and anchor matching methods that incorporate NER. This algorithm combines the speed of length-based models with the accuracy of the anchor-finding methods. Our algorithm here exploits the fact that NEs cannot be ignored from any translation process, so a sentence and its translation equivalent contain the same NEs. With NER the problem of cognate low hits for the Hungarian-English language pair can be resolved.

As input the sentence alignment method has two texts, a Hungarian and its translation in English. In the first step the texts will be sentence segmented, and then paragraph aligned. We look for the best possible alignment within each paragraph. For each possible Hungarian-English alignment we determine the cost. At each step we know the cost of the previous alignment path, and the cost of the next step can be calculated via the length-based method and anchors including NEs for each possible alignment originating from the current point (from one-to-one up to three-to-three). The base cost of an alignment is the sentence-length-difference-based one, which is increased by punishing many-to-many alignments. Without this punishment factor the algorithm would easily choose, for example, a two-to-two alignment instead of the correct two consecutive one-to-one alignments. This base cost is then modified by the matched anchors. The normalized form of the numbers, the special characters collected from the current sentences and each matching anchor together reduce the base cost by 10% and it is also reduced by 10% if the sentences have the same number of NEs.

The problem of finding the path with minimal cost (after the cost of each possible step has been determined) is solved by dynamic programming. The search starts from the first sentences of the two languages and must terminate in the final sentence of each language text. For this we used the well-known forward-backward method in dynamic programming.

5.3.3 NER results on general texts

The NER systems trained on the SzegedNE corpus and the CoNLL-2003 corpus (see sections 2.3.2 and 2.3.1) for Hungarian and English respectively were employed for the texts of the parallel corpus. We did not spend time on manually and directly evaluating Hungarian and English NER on these texts because our main goal here was the improvement of the sentence alignment system. Hence we evaluates it indirectly as the added value in the sentence alignment task.

We used the manually aligned[2] Hungarian-English parallel corpus for the experiments. This corpus contained texts taken from several sources which were quite far from the topics of business newswire NER training corpora:

[2]The work was done at the University of Szeged, Department of Informatics.

- language book sentences,
- texts gathered from an official EU website http://europa.eu under the title *Europe in 12 lessons*,
- bilingual articles taken from the travel magazines of Malév Horizon and Máv Intercity,
- Multext-East speech corpus which include topics written in everyday parlance, tells one how to order a taxi, find a restaurant, or call a customer service end so on.

The numbers of different alignment types in the output of our method and the number of correctly aligned pairs are shown in Table 5.4.

	1:1	1:2&2:1	2:2	N:M
suggested alignment	4957	339	3	1
correct of sugg. align.	4698	252	1	0

Table 5.4: The results of the sentence alignment algorithm.

To gain an insight into the usefulness of the NER system in sentence alignment we compared a hybrid alignment approach [96] without an NE-constraint and our alignment method (Table 5.5). Here, precision and recall are defined based on the number of alignments.

	Precision	Recall	$F_{\beta=1}$
hybrid algorithm	90.16	90.16	90.16
NE-extended hybrid	93.41	94.56	93.98

Table 5.5: Results of the NE-extended hybrid method.

Instead of counting the capital words taking into account the automatically recognised NEs yielded a 4% better $F_{\beta=1}$ score, which correspondes to an error reduction of 38.8%. The recognition of NEs was far from perfect, however, the obvious reason for this being the different characteristics (topics, entity types, linguistic structures etc.) of the training corpora and the texts of the parallel corpora. Even so, our NER model performed fairly well and produced a significant improvement in the performance of the sentence alignment system.

5.4 The difference between biological and medical texts

We made the first steps toward automatic domain adaptation by gathering statistics about the differences among domains, focusing on the IE tasks of negation and uncer-

tainty assertions. These tasks are essential in most IE applications where, in general, the aim is to derive factual knowledge from textual data. Take, for example, the clinical coding of medical reports, where the coding of a negative or uncertain disease diagnosis may result in an over-coding financial penalty. Another example from the biological domain is interaction extraction, where the aim is to mine text evidence for biological entities with certain relations between them. Here, while an uncertain relation or the non-existence of a relation might be of some interest for an end-user as well, such information must not be confused with real textual evidence (reliable information).

To aid the development of uncertainty and negation detection systems we built the BioScope corpus [6], which is accessible for academic purposes and is free of charge[3]. The corpus consists of three parts, namely medical free texts, biological full papers and biological scientific abstracts; and it contains annotations at the token-level for negative and speculative keywords and at the sentence-level for their linguistic scope. A module which detects uncertainty and negation along with their scope can be used as a preprocessing tool, i.e. each word in a detected scope can be removed from the documents if we seek to extract true assertions. This can significantly reduce the level of noise for processing in the kind of cases where only a document-level labeling is provided (like that for the ICD-9 coding task in Section 2.3.6) and just clear textual evidence for certain things should be extracted. On the other hand, similar systems can classify previously extracted statements based on their certainty or uncertainty, which is generally an important issue in the automatic processing of scientific texts.

Table 5.6 summarises the chief characteristics of the three subcorpora. The 3rd and 5th rows of the table show the ratio of sentences which contain negated or uncertain statements. The 4rd and 6th rows show the number of negation and hedge cue occurrences in the given corpus.

	Clinical	Full Paper	Abstract
#Documents	1954	9	1273
#Sentences	6383	2670	11871
Negation sentences	13.55%	12.70%	13.45%
#Negation cues	877	389	1848
Hedge sentences	13.39%	19.44%	17.70%
#Hedge cues	1189	714	2769

Table 5.6: Statistics of the BioScope subcorpora.

The usual language of clinical documents makes it much easier to detect negation and uncertainty cues than in scientific texts because of the very high ratio of the actual cue words (i.e. low ambiguity level), which explains the high accuracy scores reported in the literature. In scientific texts – which are nowadays becoming a popular

[3]www.inf.u-szeged.hu/rgai/bioscope

target for Text Mining (for literature-based knowledge discovery) – the detection and scope resolution of negation and uncertainty is, on the other hand, a problem of great complexity, with the percentage of non-hedge occurrences being as high as 90% for some hedge cue candidates in biological paper abstracts. Take, for example, the keyword *or*, which is labeled as a speculative keyword in only 8.85% of the cases in scientific abstracts, while it was labeled as speculative in 98.08% of the cases in clinical texts. Identifying the scope is also more difficult in scientific texts where the average sentence length is much longer than in clinical data, and the style of the texts is also more literary in the former case. We also found that hedge detection is a more difficult problem than identifying negations because the number of possible cue words is higher and the ratio of real cues is significantly lower in the case of speculation (higher keyword/non-keyword ambiguity).

These statistical figures tell us that the same tasks on slightly different domains (clinical records, full biological papers and biological abstracts) or very similar tasks on the same domain (hedge and negation detection) can have different characteristics, thus the adaptation procedure here is not straightforward.

5.5 Related work

Quite a few studies have been done on the language (in)dependence of NLP tasks (e.g. [97, 98]). The most important – from an application point of view – are the CoNLL shared tasks. More than a dozen systems participated in the multilingual NER tasks in 2002 and 2003 [82, 22], 19 and 23 predictions were submitted to the multilingual *dependency parsing* tasks in 2006 and 2007, respectively [99, 100] and they are organising the multilingual *syntactic and semantic dependency identification* task in 2009[4]. The main conclusions of these evaluation campaigns are that at this level of computational linguistic parsing, the Machine Learning models built on the primary language (usually English) can be applied without major modifications to other languages and can achieve respectable accuracy scores.

Several de-identification approaches were presented prior to the I2B2 shared task. These were based either on a pattern-matching algorithm that uses a thesaurus [101, 102]; a combination of rule-based systems and pattern matching using dictionaries [103] and the Unified Medical Language System[5] [104], or on a statistical model [105]). The participants of the first I2B2 de-identification shared task submitted both rule-based [106] and statistical approaches. The best performing systems used CRF [107, 108], AdaBoost and the C4.5 decision tree (presented here) and SVM [109, 110]. We should mention here that the best system – according to the official evaluation of the shared

[4] http://ufal.mff.cuni.cz/conll2009-st/
[5] http://www.nlm.nih.gov/research/umls/

task – adapted a general NER system to the medical domain as well.

There are quite of lot of articles available on sentence alignment (e.g. [91, 92, 93, 95]). Methods have been published for the Hungarian-English language pair by Pohl [96] and Varga et al. [111]. These are also hybrid methods that use a length-based model, but to increase the accuracy Pohl used an anchor-finding method and the algorithm developed by Varga (called Hunalign) based on a word-translation approach. Our NE-based sentence alignment method achieved significantly better accuracies compared to [96] and gave similar results to those reported in [111], but our method is significantly faster because of its accurate and fast NE detection. To the best of our knowledge our work is – among approaches for any language pair – the first sentence alignment method that uses NEs as anchors.

Lastly, as for biomedical IE there are several papers available on negation detection in medical texts (e.g. NegEx [112], NegFinder [113], MedLEE [114] and [115, 116]) but none for biological texts. As speculation detection is more important in the scientific literature there are several papers available on biological hedge detection, but they work just at the sentence-level [117, 118, 119]. Our corpus is the first with an annotation of both negative/speculative keywords and their scope, but there are biological corpora which contain negation and/or uncertainty annotation on biological events [119, 120, 121]. Although the biological and medical domains are quite similar to each other, our corpus statistics are the first comparative and empirical results (a narrative comparison was carried out in [122]).

5.6 Summary of thesis results

Our NER adaptation results between languages and domains (from newswire to medical) were introduced in the first part of the chapter. For these a training dataset was presented for the new domain and we demonstrated that the Machine Learning model with minor domain-specific modifications can be applied successfully. Then we showed that the NER models trained on business newswire texts can be applied – without a particular domain training set – to general texts and achieved satisfactory results (which were significantly improved a sentence alignment system). In the last section, we statistically investigated the difference between the biological and medical domains, which are undoubtedly the first steps toward automatic domain-adaptation.

In particular, the author made major contributions in the design, development and experimental investigations of the machine-learning-based language-independent (the system works well for different languages without the need to modify the model itself) NER system [4], which achieved competitive results with the state-of-the-art methods. Together with his colleagues, the author participated in the 2006 I2B2 shared task challenge on medical record de-identification [5]. 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 clearly show, the system we obtained via the domain adaptation of our newswire NER model is competitive with other approaches and achieved the second best scores in token-level evaluation and the best scores in phrase-level evaluation among the systems submitted to the challenge, without any statistically significant difference in performance from the other top-performing systems.

In [90] the author's contribution is the general idea of using NER in sentence alignment systems, while the other general concepts of sentence segmentation and alignment were actually carried out by the co-authors. These results show that our multilingual NER system can achieve good results on unseen domains and its use provides a way of finding appropriate anchors for language pairs even when they belong to distinct language families. In [6] the author just performed several statistical investigations, while the construction process of the BioScope corpus was carried out by the co-authors of the paper.

The key results of this chapter might seem a bit strange at a first glance and can be summarised as follows. In the middle application layer of IE systems like NER – where the deeply language-specific facts (like morphological and POS codes) are encoded into features and training sets are available – the statistical systems work language independently, while changing the domain in a certain language requires much more effort to achieve the satisfactory level of performance.

Part II

Exploitation of external knowledge in Information Extraction tasks

Chapter 6

Using the WWW as the unlabeled corpus

The most widely used external resources in an Machine Learning task are unlabeled instances. The way of training model by using unlabeled data, together with labeled data is called *semi-supervised learning*. Here the goal is to utilise the unlabeled data during the training on labeled ones. We shall give a brief overview on semi-supervised learning from a Machine Learning point of view and from an NLP point of view then we will introduce our NER refinement and lemmatisation approaches which exploit the largest unlabeled corpus in the world, that of the WWW.

6.1 Semi-supervised algorithms in Machine Learning

We classify the semi-supervised approaches into three categories, namely *generative model*s, *bootstrapping method*s and *low density separation*. For a detailed description, see [123].

The first attempts were made by applying generative models (like HMMs) [62]. A generative model directly describes how the labels are probabilistically conditioned on the inputs (tokens). 'Directly' means here that the types of distributions are assumed, and that their parameters are estimated from the data. Usually a mixture distribution is assumed and the great amount of unlabeled data helps one to identify the mixture components [62]. We regard the cluster-and-label methods (which first cluster the whole dataset, then assign a label to each cluster according to labeled data) to the generative models as well, because they perform well just via a clustering algorithm that matches the true data distribution. However, there are several problems associated with using generative models. The most obvious one is that we have to know the types of the distributions, otherwise unlabeled data reduce the accuracy [124].

We call bootstrapping methods (self-training and co-training) those type of methods where a training dataset is expanded by automatically labeled (originally) raw data [74]. Expansion means training a classifier on the actual labeled dataset and adding the most reliable examples to the training set with the predicted class label. This procedure is repeated until convergence.

The latest approaches of semi-supervised learning are based on the 'separate only on low density regions' principle (low density separation). These approaches also use the evaluation dataset as unlabeled data, hence here the overall goal is to give predictions on a specific evaluation set (transductive learning). Transductive Support Vector Machines (TSVM) [125] is an extension of SVM where unlabeled points are used to find the maximum margin linear boundary in the Reproducing Kernel Hilbert Space. In the optimisation procedure, it looks for a labeling of the unlabeled data where the margin is maximised on both originally labeled and (currently labeled) unlabeled data. TSVM is the best known low density separation method.

Graph-based methods are a newer field of low density separation [126]. Here a graph is built where nodes are labeled and unlabeled inputs and the edges represent their similarity (usually a full graph is used). If two points are in the same cluster there exists a path between them that only goes through high density regions. Thus our aim here is to learn a function (find the clusters) which cuts on low similarity edges.

The low density separation methods have a good theoretical foundations, but currently they can handle just small datasets in practice. Even packages describing themselves as solution to large-scale problems cannot give results for a task of 20,000 samples with 120 features after running for a week[1]. There are several suggestions on how to scale up these methods, but databases containing hundreds of thousands of examples as in most of the NLP tasks seem feasible only in the future.

6.2 Semi-supervision in Natural Language Processing

The special nature of NLP problems requires special semi-supervised techniques. The potential of this field has not yet been satisfactorily exploited. Two key points will be discussed here. Firstly, complex statistics can be simply gathered from unlabeled texts owing to the sequential structure of languages. Such statistics can be word and character bi-, trigrams, token or phrase frequencies and models of language in a wider sense (not just the usual $P(w_t|w_{t-1})$ distribution). This kind of information can be incorporated into the feature space for each Machine Learning process.

[1]The author downloaded and tested two packages :
www.kyb.tuebingen.mpg.de/people/fabee/universvm.html
and www.learning-from-data.com/te-ming/semil.htm

Second, a unique characteristic of NLP applications is that they can utilise the World Wide Web (WWW). The WWW can be viewed as an almost limitless collection of unlabeled data. Moreover it can bring some dynamism to applications, as online data changes and rapidly expands with time, a system can remain up-to-date and extend its knowledge without the need for fine tuning, or any human intervention (like retraining on up-to-date data, for example). On the other hand it cannot be handled by the classical semi-supervised (or unsupervised) techniques. It is feasible just via search engines (e.g. we cannot iterate through all of the occurrences of a word). There are two interesting problems here: first, appropriate queries must be sent to a search engine; second the response of the engine offers several opportunities (result frequencies, snippets, etc.) in addition to simply "reading" the pages found.

6.3 NER refinement using the WWW

During an analysis of the errors made by our NER system (introduced in Section 4.5), we discovered that a significant proportion of errors came from the machine's lack of access to the human common knowledge. If it possed such knowledge the system could not make errors like tagging the phrases *'In New York'* or give a *location* label to *'Real Madrid'*. We shall introduce WWW-based post processing techniques in order to refine the labeling of our NER model.

6.3.1 NE features from the Web

Before introducing the WWW-based post-processing techniques, we should note that our NE feature set introduced earlier (Section 3.1.4) already contains two groups of features which came from the Web. These kinds of features are frequency information and various dictionaries. The former one was gathered from corpora containing several billion tokens (Gigaword [127] and Szószablya [46]). The Named Entity dictionaries were collected from the Web as well. These lists (for a certain category) can be gathered by automatic methods via search engines and simple frame-matching algorithms [128] or parsing HTML itemisations [129], but the basic lists can be downloaded and just their filtering and normalization have to be done. The lists used in our feature set are downloaded and cleaned manually, which required less than 1 person day.

The 4-4 curves of Figure 6.1 represents the results of using the entire feature space (continuous), without frequency information (dotted), without dictionaries (dashed) and without either (longdashed). Here, the following tendency can be observed: the absence of dictionaries causes smaller loss in accuracy when the training set grows in size. The added value of dictionaries is important when only a small labeled database is present, but this information can be gained from a big labeled dataset. The use of

frequency information eliminates 19% of the errors, the dictionaries eliminate 15% and their combined usage eliminate 28% of errors [54].

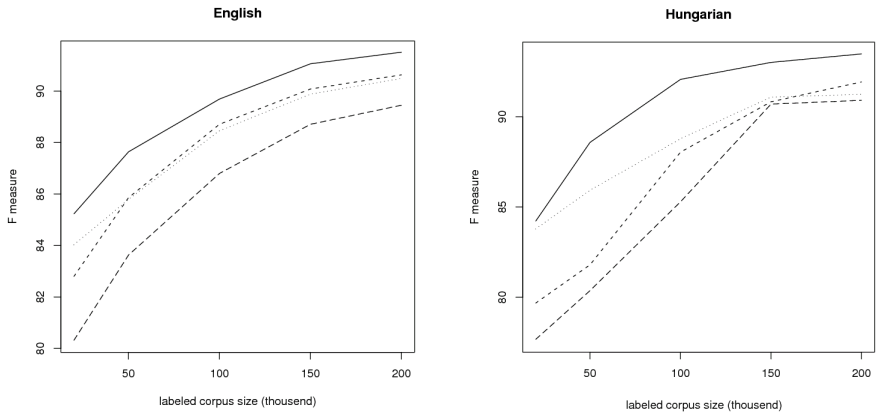

Figure 6.1: The added value of the frequency and dictionary features.

6.3.2 Using the most frequent role in uncertain cases

Some examples are easier to classify for a given model than others. In our applied NER system, the final decision was obtained by applying the majority voting procedure of 5 classifiers which were all trained on different sets of features (see Section 4.5). A simple way of interpreting the uncertainty of a decision is to measure the level of disagreement among the individual models. We considered a token as a difficult or uncertain example if no more than 2 models gave coinciding decisions (we should mention here that each models chose the most probable of 5 different possible answers, so this indeed meant a high level of uncertainty).

Our hypothesis here was that the most frequent role of a NE (the common human knowledge) can be statistically useful information. Thus we did the following: if the system was unable to decide the class label of a phrase (it could not find evidence in the context of the certain phrase) then we mined the most frequent use of the corresponding NE using the WWW and took that as prediction [7].

The most frequent role searching method we applied here was inspired by the category extraction methods of Hearst et al. [130]. This approach works by gathering such noun phrases following or preceding the pattern that is a category name for a particular class. Table 6.1 lists the queries used to obtain category names from web search results.

Category names from the training data. We used the lists of unambiguous NEs collected from the training data to acquire common NE category names. We sent Google queries for NEs in the training data and all the patterns shown above.

NP such as NE
NP including NE
NP especially NE
NE is a NP
NE is the NP
NE and other NP
NE or other NP

Table 6.1: Web queries for obtaining category names.

The heads of the corresponding NPs were extracted from the snippets of the best ten Google responses.

We found 173 reliable category names by performing a limited number of Google queries. Using these category lists as a disambiguator (we assigned the class sharing the most words in common with those extracted for the given NE) when the NER system was unable to give a reliable prediction was beneficial to the overall system performance. The system F-measure improved from 89.02% to 89.28%. We should add here that the baseline NER system labeled these examples as non-entities, whose prediction was incorrect in the majority of the cases.

Enriching category lists using WordNet. We enlisted the help of a linguist expert to determine the WordNet [131] synset corresponding to each category name we found and give its most common substituting synset (the one highest in hypo/hypernym hierarchy) that was still usable as a category name for the particular NE class. Using these WordNet synsets we extended our category lists (to a size of 19537) with every literal that appeared in their hyponym subtree (with sense #1). This additional knowledge further improved the F-measure of the NER system to 89.35%.

6.3.3 Extending phrase boundaries

A significant part of system errors in NER taggers is caused by the erroneous identification of the beginning or the end of a longer phrase. Token-level classifiers – like the one we applied for NER – are especially prone to this as they classify each token of a phrase separately.

We considered a tagged entity as a candidate long-phrase NE if it was followed or preceded by a non-tagged uppercase word, or one/two stop words and an uppercase word [7]. The underlying hypothesis of this heuristic is that if the boundaries were marked correctly and the surrounding words are not part of the entity, then the number of web-search results for the longer query should be significantly lower (the NE is followed by the particular word in just certain contexts). But in the case of a dislocated phrase boundary, the number of search results for the extended form must be comparable to the results for the shorter phrase (over 0.1% of it). This means

that every time when we found a tagged phrase that received more than 0.1% web query hits in an extended form, we extended the phrase with its neighbouring word (or words). This decision function was fine-tuned and found to be optimal on the training and development sets of the CoNLL task, and achieved a 0.13% improvement on the CoNLL evaluation set.

We came up against two problems when adapting our WWW-based approaches to the Hungarian task. First, we could not use each query of the most frequent role heuristics translated from English as the substantive verb in the third person singular is not present in Hungarian. We had to look for new query expressions and found one that was helpful: *NE egyike NP* (*NE is one of NP*). Second, the Hungarian web (we used the *site:.hu* expression in our queries) seems to be too small to get really useful responses. On average about 70% of our queries got zero results from the Google API. This fact suggests that the above mentioned WWW-based methods probably cannot provide satisfactory results for less common languages like Hungarian.

6.3.4 Separation of NEs

We examined the case where a false labeling can be corrected by extending the phrase in the previous section. Here we describe the complementary case i.e. where the separation of a labeled NE must be performed.

When two NEs of the same type follow each other, they are usually separated by a punctuation mark (e.g. a comma). Thus, if present, the punctuation mark signals the boundary between the two NEs (e.g. *Arsenal-Manchester final*; *Obama-Clinton debate*; *Budapest-Bécs marathon*). However, the assumption that punctuation marks are constant markers of boundaries between consecutive NEs and that the absence of punctuation marks indicates a single (longer) name phrase often fails (which is the case in free word order languages), and thus a more sophisticated solution is necessary to locate NE phrase boundaries.

Counterexamples for the naive assumption are NEs such as the *Saxon-Coburg-Gotha family*, where the hyphens occur within the NE, and sentences such as *"Gyurcsány Orbán gazdaságpolitikájáról mondott véleményt"*. (*'Gyurcsány expressed his views on Orbán's economic policy'* (two consecutive entities) as opposed to *'He expressed his views on Gyurcsány Orbán's economic policy'* (one single two-token-long entity)). Without background knowledge of the participants in the present-day political sphere in Hungary, the separation of the above two NEs would pose a problem. Actually, the first rendition of the Hungarian sentence conveys the true, intended meaning; that is, the two NEs are correctly separated. As for the second version, the NEs are not separated and are treated as a two-token-long entity. In Hungarian, however, a phrase like *"Gyurcsány Orbán"* could be a perfect full name, *Gyurcsány* being a family name and *Orbán* being

– in this case – the first name.

As consecutive NEs without punctuation marks appear frequently in Hungarian due to the free word-order, we decided to construct a corpus of negative and positive cases for Hungarian. Still, such cases can occur just as the consequence of a spelling error in English. Hence we focused on special punctuation marks which can separate entities in several cases (*Obama-Clinton debate*) but are part of the entity in others. In this task there are several cases where more than one cut is possible. For example, in the case of Stratford-upon-Avon, the possibilities Stratford + upon + Avon (3 lemmas), Stratford-upon + Avon, Stratford + upon-Avon (2 lemmas), Stratford-upon-Avon (1 lemma) are produced. In such cases we asked a linguist expert to choose the correct cut and every incorrect cut became a negative example. The corpora contains real-world and "interesting" cases and have a size of 200 and 100 phrases for Hungarian and English, respectively [8].

We trained and experimentally compared several classifiers on the two corpora. Such a classification system can be used as a post-processing tool for NER systems when it investigates each labeled NE phrase which contains a possible separation character (space, comma etc.) and cuts it if the classification system makes that decision. The feature set for this task is based on the queries sent to the Google and Yahoo search engines using their APIs[2]. The queries started and finished in quotation marks and the site:.hu constraint was used in the Hungarian experiments. The feature set contains six features. Two stand for the number of Google hits for the potential first and second parts of a cut and one shows the number of Google hits for the whole observed phrase. The remaining three features convey the same information, but here we used the Yahoo search engine. The two tasks are essentially binary classification problems.

We used 10-fold-cross-validation and classification accuracy as the evaluation metric in the experiments here. The baseline method classifies each sample using the most frequent label observed on the training dataset. We compared C4.5 and Logistic Regression which hade been applied successfully in classification tasks (see Section 3.1.6) and then applied the *k-Nearest Neighbour (kNN)* [132] method which is a good candidate for small sized datasets. kNN is a so-called *lazy learning* algorithms; it classifies an instance by taking the majority vote of its k nearest neighbours. Here k is a positive integer (typically not very large) and the distance among instances is usually measured by the Euclidian distance.

Table 6.2 summarises the results achieved by the classification algorithms. The Hungarian task proved a difficult one. The decision tree (which we found to be the best solution) is a one-level high tree with a split. This can be interpreted as "*if one of the resulting parts' frequency ratio is high, then it is an appropriate cut*". It is

[2]Google API: http://code.google.com/apis/soapsearch/
Yahoo API: http://developer.yahoo.com/search/

	kNN k=3	kNN k=5	C4.5	LogReg	Baseline
English	88.23	84.71	95.29	77.65	60.00
Hungarian	79.25	81.13	80.66	70.31	64.63

Table 6.2: Separation results obtained from applying different learning methods.

interesting that the learned rules for the English separation task contain constraints for the second resulting part of the separations and not just for "*one of the resulting part*"-type constraints. We guess that the bad performance of the Logistic Regression method is due to the sparse number of training samples. This amount of training instances was not enough to adequately estimate the conditional probabilities of this method.

6.4 NE lemmatisation

Finding the lemma (and inflectional affixes) of a proper noun can be useful for several reasons: the proper name can be stored in a normalised form (e.g. for indexing) and it may prove to be easier to classify a proper name in the corresponding NE category using its lemma than the affixed form. The inflectional affixes themselves can contain useful information for some specific tasks and thus can be used as features for classification (e.g. the plural form of an organisation name is indicative of org-for-product metonymies and is a strong feature for proper name metonymy resolution [3]).

Lemmatisation is not a trivial issue in morphologically rich languages. In agglutinative languages such as Hungarian, a noun can have hundreds of different forms owing to its grammatical number, possession marking and a score of grammatical cases: e.g. a Hungarian noun has 268 different possible forms [133]. On the other hand, there are lemmas that end in an apparent suffix (which is obviously part of the lemma), thus sometimes it is not clear what belongs to the lemma and what functions as a suffix.

The lemmatisation of common nouns can be made easier by relying on a good dictionary of lemmas [134]. The problem of proper name lemmatisation is more complicated since NEs cannot be listed exhaustively, unlike common nouns, due to their diversity and increasing number. Moreover, NEs can consist of several tokens (in contrast to common nouns) and the whole phrase must be taken into account. Lots of suffixes can be added to them in Hungarian (e.g. *Invitelben*, where *Invitel* is the lemma and *-ben* means '*in*' or *Pannon*, with *-on* meaning '*on*')), and they can bear the plural or genitive marker *-s* or *'s* in English (e.g. *Toyotas*). What is more, there are NEs that end in an apparent suffix (such as *Adidas*, *McDonald's* or *Philips*), but this pseudo-suffix belongs to the lemma of the NE and should not to be removed.

In order to be able to select the appropriate lemma for each problematic NE, we

applied the following strategy. In step-by-step fashion, each ending that seems to be a possible suffix is cut off the NE. Our key hypothesis is that the frequency of the lemma-candidates on the WWW is high – or at least the ratio of the full form and lemma-candidate frequencies is relatively high – with an appropriate lemma and low in incorrect cases.

In order to verify our hypothesis we manually constructed two corpora of positive and negative examples for NE lemmatisation. 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. 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, The Hungarian and English corpora contained 750 and 160 examples and are accessible free of charge.

We sent queries with and without suffixes to Google and Yahoo search engines and collected the number of hits. The original database contained four dimensional feature vectors. Two dimensions listed the number of Google hits and two components listed similar values from the Yahoo search engine. We again trained the kNN (with $k = 3$), C4.5 and Logistic Regression classifiers to find the decision boundary for appropriate lemmas based on the frequency results of the search engines.

Our preliminary experiments showed that using just the original form of the datasets ("orig." rows of Table 6.3) is not optimal in terms of classification accuracy. Hence we performed some basic transformations on the original data, the first component of the feature vector being divided by the second component ("rate" rows in Table 6.3). If the given second component was zero, then the new feature value was also zero. This yielded a two dimensional dataset when we utilised both Yahoo and Google hits for the suffix classification tasks and a four dimensional for the separation tasks. Finally, we took the minimum and the maximum of the separated parts' ratios, which provided the possibility to learn rules such as *'if one of the separated parts' frequency ratio is higher than X'*.

The size of the English database is quite small, which leads to a dominance of the k-Nearest Neighbour classifier, since the lazy learners usually achieve good results on small datasets. 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 hits (one feature) is the best characterisation in this task. However, we can see that with the Hungarian dataset the original dataset characterises the problem better, thus the transformation is really

	Search Engine	Features	kNN	C4.5	LogReg	Baseline
English	Google	orig.	87.34	87.34	82.28	53.16
		rate	93.67	87.97	90.51	53.16
	Yahoo	orig.	89.87	86.08	84.18	53.16
		rate	91.77	87.34	88.61	53.16
	Both	orig.	89.24	86.71	84.81	53.16
		rate	93.04	91.77	73.42	53.16
Hungarian	Google	orig.	85.33	82.40	83.33	72.40
		rate	83.60	83.60	77.60	72.40
	Yahoo	orig.	93.73	83.87	86.13	72.40
		rate	87.20	87.20	74.00	72.40
	Both	orig.	94.27	82.67	88.27	72.40
		rate	84.67	81.73	72.40	72.40

Table 6.3: Lemmatisation results achieved by different learning methods.

unnecessary.

The best results for the Hungarian lemmatisation task are achieved on the full dataset, but they are almost the same as those for the 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.

Despite the small size of our corpora, we got fairly good empirical results (a 91.09% average accuracy) and we expect even better accuracy can be obtained with bigger corpora.

6.5 Related work

There are several papers in the literature that investigate the usability of online resources for various NE-related tasks. The available systems seek to collect lists of Named Entities belonging to pre-specified classes from the WWW [128, 135, 129] or use online information for Named Entity Disambiguation [136, 137]. We found no articles on using Web-searches to post-process a NER system.

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 method and several string distance-based methods for Polish person name inflection removal were introduced in [138]. A corpus-based rule induction method was studied for every kind of unknown words in Slovene in [139]. The scope of our work lies between these two as we deal with different kinds of NEs. To best of our knowledge, our approach was the first attempt at Web-based NE lemmatisation.

6.6 Summary of thesis results

The main results of this chapter is to highlight the exploitation potential of the WWW – the largest external resource available in the world – in NLP solutions like NER. The heuristics introduced are based on the assumption that, even though the World Wide Web contains a good deal of useless and incorrect information, for our simple features the frequency of correct language usage dominates misspellings and other sorts of noise.

The author with his colleagues developed WWW-based NER post-processing heuristics and experimentally investigated them on general reference NE corpora [7]. They constructed several corpora for the English and Hungarian NE lemmatisation and separation tasks [8]. The NE lemmatisation task is important for textual data indexing systems, for instance, and is of great importance for agglutinative languages like Finno-Ugric and Slavic languages. Based on these constructed corpora, automatically derived simple decision rules were introduced. Experiments confirmed that the result frequencies of search engines provide enough information to support such NE related tasks.

The author's own contributions in the Web-based solutions are the most frequent role and phrase extension approaches in [7], while the feature engineering tasks and most of the Machine Learning experiments of [8] were carried out by the author.

Chapter 7

Integrating expert systems into Machine Learning models

Besides unlabeled texts, existing expert decision systems, manually built taxonomies or written descriptions can hold useful information about the Information Extraction task in question. A fine example for this is clinical IE where the knowledge of thousands years is gathered into medical lexicons. On the other hand hospitals and clinics usually store a considerable amount of information (patient data) as free text, hence NLP systems have a great potential in aiding clinical research due to their capability to process large document repositories both cost and time efficiently. We shall introduce several ways of integrating the medical lexical knowledge into Machine Learning models – which were trained on free-text corpora – through two clinical IE applications, namely *ICD coding* and *obesity detection*.

7.1 Automated ICD coding of medical records

We built an automated ICD coder on the CMC dataset (see Section 2.3.6) which exploited the existing coding knowledge (present in the coding guidelines) while training Machine Learning models on a labeled corpus [9].

7.1.1 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 [140], [141] and [142]. 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 7.1 shows how expert rules are generated from an ICD-9-CM coding guide. The system of Goldstein et al. [143] applies a similar approach and incorporates knowledge from [142].

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

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. The two most important points which have to be dealt with to get a high performance coding system are the lack of coverage of the source dictionary (missing synonyms or phrases that appear in real texts) and the lack of knowledge about inter-label dependencies needed to remove related symptoms when the code of a disease is added.

7.1.2 Language pre-processing for the statistical approach

In order to perform the code classification task accurately, some pre-processing steps have to be performed to convert the text into a consistent form and remove certain parts. First, we lemmatized and converted the whole text to lowercase (we used the freely available Dragon Toolkit [144]). Next, the language phenomena – negation and speculation – that had a direct effect on ICD-9-CM coding were dealt with. As a final step, we removed all punctuation marks from the text.

According to the official coding guidelines, negated and speculative assertions (also referred to as soft negations) have to be removed from the text as negative or uncertain diagnosis should not be coded in any case. We used the punctuations in the text to determine the scope of keywords. We identified the scope of negation and speculative keywords to be each subsequent token in the sentence. For a very few specific keywords

(like *or*) we used a left scope as well, that was each token between the left-nearest punctuation mark and the keyword itself. We deleted every token from the text that was found to be in the scope of a speculative or negation keyword prior to the ICD-9-CM coding process. Our simple algorithm is similar to NegEx [112] 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.

In our experiments we found that a slight improvement on both the training and test sets could be achieved by classifying the speculative parts of the document in cases where the predicative texts were insufficient to assign any code. This observation suggests that human annotators tend to code uncertain diagnosis in those cases where they find no clear evidence of any code (they avoid leaving a document blank). Certainly, negative parts of the text were detrimental to accuracy in any case.

We made use of negation and speculative keywords collected manually from the training dataset. Speculative keywords which indicate an uncertain diagnosis were collected from the training corpus: *and/or, can, consistent, could, either, evaluate, favor, likely, may, might, most, or, possibility, possible, possibly, presume, probable, probably, question, questionable, rule, should, sometimes, suggest, suggestion, suggestive, suspect, unless, unsure, will, would*.

Negation keywords that falsify the presence of a disease/symptom were also collected from the training dataset: *cannot, no, not, vs, versus, without*.

The accurate handling of these two phenomena proved to be very important on the challenge dataset. Without the negation filter, the performance (of our best system) decreased by 10.66%, while without speculation filtering the performance dropped by 9.61%. We observed that there was a 18.56% drop when both phenomena were ignored.

The above-mentioned language processing approach was used throughout our experiments to permit a fair comparison of different systems (each system had the same advantages of proper preprocessing and the same disadvantages from preprocessing errors). As regards its performance on the training data, our method seemed to be acceptably accurate. On the other hand, the more accurate identification of the scope of keywords is a straightforward way of further improving our systems.

Example input/output pairs of our negation and speculation handling algorithm:

1. **Input:** *History of noonan's syndrome. The study is being performed to* ***evaluate*** *for evidence of renal cysts.*

   Output: *History of noonan's syndrome. The study is being performed to.*

2. **Input:** *Mild left-sided pyelectasis,* ***without*** *cortical thinning* ***or*** *hydroureter. Normal right kidney.*

   Output: *Mild left-sided pyelectasis. Normal right kidney.*

Temporal aspects should also be handled as earlier diseases and symptoms (in case they have no direct effect on the treatment) should either not be coded or be distinguished by a separate code (like that in the case of code *599.0* which stands for *urinary tract infections*, and *V13.02* which stands for *history of urinary tract infections in the past*). Since we were unable to find any consistent use of temporality in the gold standard labeling, we decided to ignore the temporal resolution issue.

7.1.3 Multi-label classification

An interesting and important characteristic of the ICD-9-CM labeling task is that multiple labels can be assigned to a single document. Actually, 45 distinct ICD-9-CM codes appeared in the CMC Challenge dataset and these labels formed 94 different, valid combinations (sets of labels).

There are two straightforward ways of learning multi-label classification rules, namely treating valid sets of labels as single classes and building a separate hypothesis for each combination, or learning the assignment of each single label via a separate classifier and adding each predicted label to the output set. Both approaches have their advantages, but they also have certain drawbacks. Take the first one; data sparseness can affect systems more severely (as fewer examples are available with the same set of labels assigned), while the second approach can easily predict prohibited combinations of single labels.

Preliminary experiments for these two approaches were carried out: Machine Learning methods were trained on the Vector Space representation (language phenomena were handled but the ICD-9-CM guide was not used). In the first experiment we used 94 code-combinations as the target class of the prediction and we trained 45 classifiers (for each code separately) in the second one. Based on the preliminary results (see Table 7.2) we decided to treat the assignment of each label as a separate task and made the hypothesis that in an invalid combination of predicted labels any of them could be incorrect.

7.1.4 Combining expert systems and Machine Learning models

Although the expert rules contain many useful phrases that are indicators of the corresponding label with a very high confidence, the coverage of these guides is not perfect. There are expressions and abbreviations which are characteristic of the particular health institute where the document was created, and physicians regularly use a variety of abbreviations. As no coding guide is capable of listing every possible form of every concept, to discover what these infrequent keywords are an examination of labeled data is nec-

essary.

On the other hand the labeled corpus provides the opportunity of training classifiers, but they offer no chance to train rare labels because of the data spareness. After the employment of the above-mentioned language pre-processing steps, the Vector Space Model can be readily applied. We used here a token-level Vector Space representation of the documents (token uni-, bi- and trigrams) as a feature set for the statistical models.

We shall introduce three different methods to utilise the advantages of both the expert rules and the labeled data:

Extended rule-based system: We tried to solve the incompleteness of rule-based system by gathering synonyms and abbreviations from the training corpus. This extension of the synonym lists can be performed via a manual inspection of labeled examples, but this approach is most laborious and hardly feasible for hundreds or thousands of codes, or for a lot more data than in the challenge. Hence this task should be automated, if possible. The effect of enriching the vocabulary is very important. This step reduced the classification error by 30% when we built a system manually.

Since missing transliterations and synonyms can be captured through the false negative predictions of the system, we decided to build statistical models to learn to predict the false negatives of our ICD-9-CM coder. This way we expected to have the most characteristic phrases for each label among the top ranked features for a classifier model which predicted the false negatives of that label. We used the C4.5 decision tree learning algorithm for this task because it builds models that are very similar in structure to the rule-based system. With this approach we managed to extend the rule-based model for 10 out of 45 labels. About 85% of the new rules were synonyms (e.g. *Beckwith-Wiedemann syndrome, hemihypertrophy* for *759.89 Laurence-Moon-Biedl syndrome*) and the remaining 15% were abbreviations (e.g. *uti* for *599.0 urinary tract infection*).

Extended classification system: We can import the rule-based system into the classification model by incorporating its predictions into the feature space of the latter. We added all the codes predicted by the rule-based system to the Vector Space Model representation. Thus the statistical system can exploit the knowledge of both the coding guides and the regularities of the labeled data. One of the obvious drawbacks of this approach is that the classifier builds decision rules on the features based on the expert system when it sees a sufficient number of samples.

Hybrid model: As we have already mentioned, the expert system has a high precision

and a lower recall. Thus one of the most straightforward approaches for the combination in this multi-labeling environment is to take the union of the labels predicted by the rule-based expert system and the Machine Learning model. In this setting we made predictions using the expert system and the classifier quite independently.

7.1.5 Results on ICD-coding

Table 7.2 overviews the results achieved by the ICD-coding approaches introduced above [9]. All values are micro-averaged $F_{\beta=1}$ measures, the official evaluation metric of the International Challenge on Classifying Clinical Free Text Using Natural Language Processing. The *94-class statistical* row stands for the C4.5 classifier trained for code-combinations and the *45-class statistical* row stands for the same classifier trained for single labels. The *rule-based system* of the 3rd row is the original one extracted from the coding guide, while the last one (*manually built system*) stands for its manual extension by synonyms and abbreviations observed on the training data (this process required 3 days for the 45 codes).

All our models use the same algorithm to detect negation and speculative assertions, and were trained using the whole training set (as a 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 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 was not statistically significant on either set.

	train	test
94-class statistical	83.06	82.27
45-class statistical	88.20	86.69
Rule-based from coding guide	85.57	84.85
Extended rule-based	90.22	88.93
Extended classification	90.62	87.92
Hybrid model	90.53	89.33
Manually built expert system	90.02	89.08

Table 7.2: Overview of the ICD-9-CM results.

The CMC Challenge itself was dominated by entirely or partly rule-based systems that solved the coding task using a set of hand crafted expert rules. The feasibility of the construction of such systems for thousands of ICD codes is indeed questionable. Our results are very promising in the sense that we managed to achieve comparable results with purely hand-crafted ICD-9-CM classifiers. Our results demonstrate that

hand-crafted systems – which proved to be successful in ICD-9-CM coding – can be reproduced by replacing several laborious steps in their construction with statistical learning models. These hybrid systems preserve the favourable aspects of rule-based classifiers and thus achieve a good performance, and can be developed rapidly and requires less human effort. Hence the construction of such hybrid systems can be feasible for a set of labels one magnitude bigger, and with more labeled data.

7.2 Identifying morbidities in clinical texts

Classifying patient records whether they have a certain disease is a similar task to ICD coding. The *Obesity Challenge* in 2008, organised by the Informatics for Integrating Biology and the Bedside (I2B2)[1], asked participants to construct systems that could correctly replicate the textual and intuitive judgments of the medical experts on obesity and its co-morbidities based on narrative patient records [145]. The development of systems that can successfully replicate the decisions made by obesity experts would be desirable to facilitate large scale research on obesity, one of the leading preventable causes of death [146].

The target diseases included *obesity* and its 15 most frequent co-morbidities exhibited by patients, while the target labels (positive, negative, questionable, unmentioned) corresponded to expert judgements based on *textual evidence* and *intuition*. That is, for each patient, both what the text explicitly said about obesity and co-morbidities, and what the text implied about obesity and co-morbidities, were provided as gold standard labels by obesity experts. The dataset consisted of 1237 discharge summaries. Each document had been annotated for obesity and the other 15 diseases. Out of these documents, 730 were made available to the challenge participants for development and the remaining 507 documents constituted the evaluation set.

Our textual classification approach focused on the rapid development of an extended dictionary-lookup-based systems, which also took into account the document structure and the context of disease terms for classification. To achieve this, we used statistical methods to pre-select the most common (and most confident) terms and abbreviations then evaluated outlier documents to discover infrequent terms and spelling variants.

Sentences containing disease terms were then further investigated to decide whether the term was in assertion and it was relevant (information on the patient and not on family members, etc.). Keyword lists were exploited to identify negations, hedges and judging the relevance. Manually gathered delimiter lists were used to determine their scope, and terms within the scope of a negation or uncertain cue were handled using this information.

[1] www.i2b2.org/NLP/

In our intuitive model, we attempted to discriminate the documents classified as unmentioned by our textual classifier into positive or negative intuitive classes. For this task, we extended the system with *intuitive dictionaries* gathered from external sources. The phrases of these dictionaries were typically names of associated drugs and medication, or phrases related to certain social habits of the patients (e.g. cigarette for hypertension). The chief source of the dictionaries were the MedlinePlus encyclopedia[2]. Lists of phrases were downloaded and then filtered for intuitive positive class-conditional probability. In this case the combination of the data-driven model and the external knowledge was straightforward; the final *intuitive dictionaries* (the "expert system") were used to classify a documents as an intuitive positive when it was judged to be unmentioned by the textual classifier system.

According to the official evaluation metric of the challenge [145] – a macro-averaged F-measure among the four classes – our system achieved an F-macro of 84% on the train for our best model (which degraded to 76% on the test), and an intuitive F-macro of 82% on train (which degraded to 67% on the test). The micro-averaged results were in the high 90s. Those classes that had a few hundred training examples (positive & unmentioned for textual and positive & negative for intuitive annotation) generally achieved an F-measure of about 97%. This suggests that our approach is indeed capable of locating the most relevant pieces of information for each of the 16 diseases addressed in almost all of the documents. Lower scores were observed for infrequent classes (with only 1-10 examples on average per class/disease pair) and we think that having more examples for questionable cases and negative examples would result in a substantial improvement in performance on these particular classes as well.

7.3 Related work

The possibilities of automating the detection of diseases in textual medical records have been studied extensively since the 1990s. Larkey and Croft [147] assigned ICD 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. [148] gave an overview of the problem in a feasibility study. Lima et al. [149] 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.

The most recent results are clearly related to shared tasks in 2007 and 2008. To the 2007 CMC Challenge on Classifying Clinical Free Text (see Section 2.3.6), 44 teams submitted well-formatted results. 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

[2] http://www.nlm.nih.gov/medlineplus/encyclopedia.html

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) [150, 151, 152, 153, 143].

Our manually built system (which extended the coding guide by synonyms and abbreviations) won the challenge in 2007 [9]. We constructed it so to be a baseline system for our Machine Learning experiments but we could not outperform it during the challenge development phase. After the challenge we investigated our data-driven models in-depth and found that they could achieve a performance without any significant difference compared to our manual model, i.e. each step of the manual rule construction could be replaced by classification models trained on a hand-labelled corpus.

28 teams submitted valid predictions to the I2B2 obesity challenge in 2008. The two main approaches of participants were the construction of rule-based dictionary lookup systems and Bag-of-Words (or bi- and trigram-based) statistical classifiers. The dictionaries of the systems mostly consisted of the names of the diseases, and their various spelling variants, abbreviations, etc. One team also used other related clinical named entities [154]. The dictionaries employed were constructed mainly manually (either by domain experts [155] or computer scientists [156]), but one team applied a fully automatic approach to construct their lexicons [157]. Machine learning methods applied by participating systems ranged from Maximum Entropy Classifiers [158] and Support Vector Machines [154] to Bayesian classifiers (Naive Bayes [159] and Bayesian Network [160]). These systems showed competitive performance on the frequent classes but had major difficulties in predicting the less represented negative and uncertain information in the texts.

Our dictionary-lookup-based system were extended by term-context analysis. The lists of phrases were resulted by the statistical filtering of external encyclopedia and the training dataset. This system required just a very rapid development time, while achieved comparable results. It came 6th in the textual F-macro ranking and 2nd in the intuitive F-macro ranking of the shared task.

7.4 Summary of thesis results

There are several tasks where expert rule-based systems are available along with manually labeled datasets. We introduced two such tasks – two clinical Information Extraction tasks – ICD coding and disease/morbidity detection in this chapter. Medical encyclopedical knowledge forms expert rule-based systems here in a straightforward way.

The author with his co-authors developed solutions for these tasks [9, 10] that integrates Machine Learning approaches and external knowledge sources. They exploited the advantages of expert systems which are able to handle rare labels effectively. Sta-

tistical 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, rarely used medical terms or other linguistic structures.

Each statistical system along with the described integration methods developed for the two tasks were the author's own contributions.

Overall, we think that our results demonstrate the real-life feasibility of our proposed approach and that even very simple systems with a shallow linguistic analysis can achieve remarkable accuracy scores for information extraction from clinical records. In a wider context, the results achieved by the participating teams (with the team 'Szeged') of the shared tasks demonstrate the potential of Natural Language Processing in the clinical domain.

Chapter 8

Exploiting non-textual relations among documents

In an applied Information Extraction task the documents to be processed are usually not independent of each other. The relation among documents – external knowledge – can be exploited in the IE task itself. We shall introduce two tasks where graphs are constructed and employed based on these relations. In the biological *Gene Name Disambiguation* task we will utilise the co-authorship graph, while in the *Opinion Mining* task the response graph will be built.

8.1 Co-authorship in gene name disambiguation

Biolgical articles provide a huge amount of information about genes, proteins, their behaviour under different conditions, and their interactions. The handling of huge amounts of unstructured data (free text) has increased in interest along with the application of automatic NLP techniques to biomedical articles. NER is the first and crucial step of a biological Information Extraction system and a major building block of an Information Retrieval system as well.

The task of biological entity recognition is to identify and classify gene, protein, chemical names in biological articles [161]. Taken one step further, the goal of Gene Name Normalisation (GN) [32] is to assign a unique identifier to each gene name found in a text. The GN task is challenging for two main reasons. First, although synonym (alias) lists which map gene name variants to gene identifiers exist like that given in [36], they are incomplete and they do not contain all the spelling variants [162]. On the other hand one name can refer to different entities (for example, *IL-21* can refer to the genes with EntrezGeneID 27189, 50616 or 59067). Chen et al. [163] investigated gene name ambiguity in a comprehensive empirical study and reported an average of 5% overlap on intra-species synonyms, and ambiguity rates of 13.4%, and

1.1% on inter-species and against English words, respectively. In general, the Word Sense Disambiguation (WSD) approaches (for a comprehensive study, see [164]) are concerned with this crucial problem. Their goal is to select the correct sense – from a well-defined sense inventory – of a term according to its context. A special case of WSD task is the Gene Symbol Disambiguation (GSD) [33] task where the terms are gene names, the senses are genes referred by unique identifiers and the contexts are biological articles.

The datasets used in our GSD experiments were introduced in Section 2.3.7.

8.1.1 The inverse co-author graph

Our main idea [11] is that an author habitually uses gene names consistently; that is, they employ a gene name to refer exclusively to one gene in their publications. Generalising this hypothesis we may assume that the same holds true for the co-authors of the biologist in question. But what is the situation for the co-authors of the co-authors? To answer this question - and utilise the information obtained from co-authorship in the GSD problem - we decided to use the so-called co-author graph [165].

The co-author graph represents the relationship between authors. The nodes of the graph are authors, while the edges represent mutual publications. In the GSD task we basically look for an appropriate distance (or similarity) metric between pairs of abstracts, hence we define the *inverse co-author graph* as a graph whose nodes are abstracts from MedLine (we usually just used their PMID and not their actual text) and there is an undirected edge between two nodes if and only if the intersection of their author sets is not empty.

To get the inverse co-author graph we downloaded (in April of 2007) all MedLine abstracts, which contained some 11.7 million instances. We could not construct the whole graph due to space and time restrictions, but we constructed the subgraph of each test example (the dataset introduced in Section 2.3.7) surroundings (nodes reachable in five steps). The number of articles reached in 3 steps (7.2 million for human, 0.7 million for mouse, 0.7 million for yeast and 50 thousand for fly) gives an indication of the amount of studies dealing with each species in question and helps explain the difficulties we had when processing the human dataset.

8.1.2 The path between the test and train articles

In our first approach we examined how strong the co-authorship was between the test article and the train articles. The strength of the co-authorship can be measured as the distance between two nodes in the inverse co-author graph. When two nodes are neighbours the two articles have a mutual author. When a node can be reached in two steps, starting from a node means that the two articles have no mutual authors, but

some of the authors have a mutual publication (excluding the two articles in question). We looked for the shortest path from the test node to each train example in the inverse co-author graph. Among the closest training points (we gathered all training samples which had the same minimal distance) a majority voting was applied i.e. we made a disambiguation decision in favour of the gene with the closest labelled nodes. Table 8.1 lists the precision and recall values (in a precision/recall format) we obtained by this method using non-weighted path lengths. A coverage over 90% was achieved on the mouse, fly and yeast datasets by just considering the neighbours of the test nodes, which implies that test nodes and most of the train nodes have a co-author. Significantly fewer articles deal with these organisms than with human and these articles can be processed in a higher coverage by the Entrez group.

Distance Lim.	Human	Mouse	Fly	Yeast
1	100 / 44.35	99.88 / 97.59	99.84 / 92.19	100 / 99.26
2	100 / 49.19	98.67 / 99.32	94.58 / 97.72	100 / 99.26
3	85.29 / 82.26	98.64 / 99.51	94.44 / 98.10	100 / 99.26

Table 8.1: Results obtained using the path-length-based method.

In our experiments we found that if there was a path between the test node and one of the train nodes (this is true in over 90% of the cases) its length was at most 3. We did not examine this property on the complete graph, but - interpreting training and test nodes as a random sample of node pairs from the graph - we can suppose that the average minimum path length between nodes (articles) is surprisingly small (3 or 4).

8.1.3 Filtering and weighting of the graph

Table 8.1 tells us that the noise is considerable in cases where the distance between the closest training node and the test node is 3. We tried to eliminate the noise of these distant training points hence we left out the less reliable edges from the graph. Our hypothesis was that the authors who have a large number of publications do not have a bigger influence and correspondence in articles, hence the edges originating from them are less reliable. To test this hypothesis we ignored the last 10% of the authors from each article, and then repeated each experiment by ignoring those authors who had over 20, 50 or 100 MedLine publications.

We investigated two edge-weighting methods on the human dataset along with the filtering process. We calculated the weight w for each edge as a function of the number of mutual authors of the two given articles like so:

$$w = \sum \frac{2}{|A \bigcap B|} / \min(|A|, |B|),$$

where A and B are the sets of the authors of the articles. To get an aggregated, weighted distance for a path we summed the edge-weights ($D_{sum} = \sum_i w_i$) or used the minimum of the edge-weights, i.e. the bottleneck of the path ($D_{min} = \min_i w_i$).

After calculating the weighted path lengths for each train node we chose (instead of the closest training examples' majority voting) the label of the node with the maximal weight as the final disambiguation prediction.

The different degrees of filtering resulted in different precision and coverage value pairs. Figure 8.1 shows the precision-recall curves obtained using the three weighting methods (i.e. non-weighted, D_{sum} and D_{min}). The points of the curves refers to different levels of filtering of the inverse co-author graph. The authors who had over 100, 50 or 20 MedLine publications were ignored yielding 3 points on the precision-coverage space, while the fourth point of each curve shows the case without any filtering.

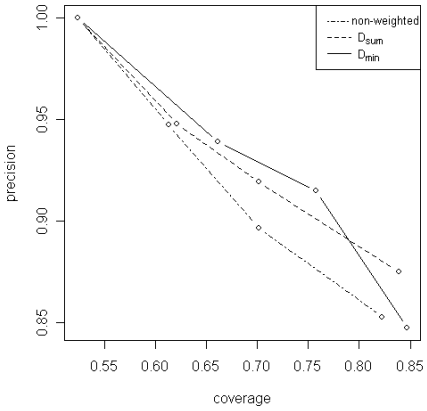

Figure 8.1: Precision-recall curves on the human GSD dataset.

According to these results, ignoring more authors from the co-author graph yields a higher precision but at the price of lower recall. Thus this filtering approach is a parametric trade-off between precision and recall. A 100% precision can be kept with a recall of 54.42% while the best coverage achieved by this method was 84.67% with a decrease in precision to 84.76%. The difference between the performance of the three weighting (or non-weighting) methods is significant. The right choice of a method can yield a 2-3% improvement in precision at a given level of recall. The minmax method seems to outperform the other two, but it does not perform well on the unfiltered graph hence we cannot regard it as the ultimate 'winning' solution here.

8.1.4 Automatic expansion of the training set

The absence (or small number) of training examples in several cases (especially on the human evaluation set) makes the GSD tasks intractable. To overcome this problem,

we extended the labelled set automatically by articles based on the inverse co-author graph. We assumed here that the probability of an author dealing with the same gene in more articles is higher than the probability of dealing with different genes which share an alias. Thus we looked for gene aliases among the articles of the authors and hoped that they used a synonym (or long form) of the target gene name. For example, *CASPASE* in *PMID:12885559* can refer to genes with EtrezGeneID *37729* or *31011* and the document does not contain any synonym belonging to them. One of the authors (*McCall K.*) has two other publications *PMID:999799* and *PMID:9422696* which contain *DCP-1* (EntrezGeneId *37729*), so we assumed that *CASPASE* refers to *DCP-1* in the test abstract. Our assumption is questionable but as our experiments show it is true in over 90% of the cases.

We labelled each article in the neighbourhood of the test node with a gene identifier if a synonym of the target gene name was found (with exact string matching) in the document. Note that the test abstract (distance 0) can also contain synonyms of the target gene name. In these cases, we made a decision based on this information as well (the special case of distance 0 is equivalent to the disambiguation procedure described in [166]).

Dist. Lim.	Human	Mouse	Fly	Yeast
0	93.30 / 12.11	96.28 / 8.57	100 / 7.06	83.70 / 10.23
1	92.56 / 32.82	91.41 / 18.82	96.56 / 10.78	69.75 / 18.79
2	91.53 / 37.88	91.31 / 20.07	96.56 / 10.78	69.75 / 18.79

Table 8.2: Results obtained using the automatic labelled set expanding heuristic.

After this expansion we made the disambiguation decision via the non-weighted majority voting method on the new set of train samples. Table 8.2 shows the precision and recall values we got with this procedure on the four datasets. These values tell us that the articles with a distance of two hardly ever contains gene aliases, which leads to a slight improvement in the recall rate. We should add that there is a strong statistical connection between the achieved recall by this method on the particular organism and the size of the available synonym list and labelled train sets.

We combined the two co-author graph-based methods (minimal path finding and training set expansion) to exploit the advantages of both via the following strategy: when there is at least one training node in the neighbourhood of distance 3 of the test node on the filtered graph, we accept the decision of that model. If there is no such close train node we try to label new documents with the synonym list and make a decision based on these automatically labelled instances. We got some results by applying two filtering and weighting procedure combinations, one yielding a maximal precision and the other a maximal recall. The precision and recall values we got of the

combined co-author-based method can be found in Table 8.3.

Method	Human	Mouse	Fly	Yeast
max precision	100 / 52.42	99.76 / 97.80	99.59 / 92.42	100 / 99.25
max recall	84.76 / 84.67	99.48 / 98.74	97.94 / 95.68	100 / 99.25

Table 8.3: Results obtained using the combined co-author-based methods.

8.1.5 Achieving the full coverage

In a real world biomedical application the aim is usually to make a disambiguation decision on every gene mention found. As the last rows of Table 8.2 and 8.3 make clear, the maximum recall which can be achieved by our best inverse co-author graph-based methods is about 85% on human (and over 98% for the other 3 species). In the last part of our experiments we investigated what effect our co-author graph-based heuristics has in a gene disambiguation system which runs on 100% coverage.

We employed two methods, namely the similarity-based procedure introduced by [167] and a supervised Machine Learning approach. In the first case we chose the gene with the maximal cosine similarity between the test article and the centroid of the training samples belonging to a given gene (gene profile). This method was used earlier by [34] and we re-implemented it for the sake of making a comparison between their approach and ours.

In the supervised learning case, information provided by MedLine were used as features including the MeSH headings (manually annotated in the MedLine) and information about the release of the articles, the journal title, and the year of publication but we did not make use of the text itself. There were several reasons for this. First of all, the manually added MeSH headings represents very well the biological concepts of the article in a normalised and disambiguated way. Second, the empirical results of [35] on two evaluation sets show that using the words of the text along with MeSH headings could not achieve any significant improvement. We also examined the potential of the combined use of headings and text (we lemmatised the text and ignored stop words) in preliminary experiments, but no significant improvement was found either hence the text itself was left out for time complexity reasons. We trained a C4.5 decision tree [38] on the feature set introduced above and accepted its forecast on the test example as a final disambiguation decision.

Table 8.4 summarises the results of these two methods applied separately and in combination with the co-author-based heuristics. In this final hybrid system we first applied these two co-author graph-based procedures with filtering to get the highest precision. Then as a second step, we applied a similarity or Machine Learning technique on the instances where the first step could not make any decision.

The first row of Table 8.4 lists the precision and recall values of a baseline method. As a standard in WSD, we used the baseline of choosing the majority sense (the gene having the most training examples) of each gene mention. We represented the results of Xu et al. by using MeSH codes in the second row for the sake of comparability. The results of a C4.5 decision tree using the MeSH features are present in the third row. The systems of the two last rows first apply the combined co-author graph-based heuristics and when they cannot decide they use the supervised prediction of the cosine similarity metric or the decision tree.

Method	Human	Mouse	Fly	Yeast
Baseline	59.3 / 99.1	79.0 / –	66.7 / –	65.5 / –
Xu et al. [34, 35] MeSH	86.3 / 94.4	90.7 / 99.4	69.4 / 99.7	78.9 / 98.4
Decision tree	84.7 / 100	90.9 / 99.8	72.5 / 99.9	74.5 / 100
Co / authorship+similarity	91.9 / 99.2	98.5 / 99.8	97.2 / 100	94.2 / 99.7
Co / authorship+decision tree	94.4 / 100	98.9 / 99.9	96.1 / 99.9	99.6 / 100

Table 8.4: Overview of GSD systems which aimed at full coverage.

From a supervised learning point of view the co-author graph-based heuristics eliminate 80% of the errors (decreasing the average error from 18.67% to 4.5% for the similarity measure and from 19.85% to 2.8% for the decision tree), while from the co-author graph point of view the doubtful examples can be predicted with an 80% precision by supervised techniques, thus yielding a full coverage with an aggregated precision of 97.22%.

8.1.6 Discussion on the GSD task

There are quite significant differences among the tasks of the given species. The human GSD evaluation set is without doubt the most difficult one for the co-authorship-based approaches because of the extremely large number of articles which focus on this organism and the relative modest number of average training samples available. The co-authorship method achieves precision values over 99% with a recall of over 92% on the other three datasets. The final results with a complex method (co-authorship-based heuristics along with supervised techniques) correlate with the baseline values (and the simple supervised methods) i.e. mouse is the best performing one and a lower precision is obtained on human and fly. The final results on yeast are surprising as baseline methods on this dataset performed the worst but achieved the best results when the co-authorship-based methods were applied (and in the final one as well). We think that this is because of the small amount of articles which focus on this organism, which might imply a smaller author society with stronger relationships.

The difference between the baselines and the purely supervised models and the difference between supervised models and final models which employ co-author graph-

based heuristics are statistically significant, due to the McNemar's test with a $p < 0.05$ confidence level, but the difference between the two supervised models was below the statistical level of significance. This holds true for the cases of their use in the final cascade systems as well. The decision tree (when sufficient amount of training data is available) can differentiate the features in a more sophisticated way than the vector space model can. Furthermore, the decision tree can learn complex rules like "the papers released before 2002 and containing Mesh code X but not containing Mesh code Y are ...". However, with these complex modeling issues it could not achieve a statistically significant difference compared to the similarity-based approach. This could be because of the small training sets and overfitting. But we suggest using decision trees because its learnt model is human readable so a domain expert can understand and modify it when necessary.

The most obvious limitation of our co-authorship-based approach is that it is dependent on a training set derived from manually disambiguated annotation by the Entrez group. On viewing Table 8.1, we see that if the number of annotated articles were higher the GSD task would become a trivial one. There are two factors of the graph construction approach which seem to be negligible but nevertheless deserve a mention here. First, an edge is drawn between nodes because of string matching of the author names. Of course, the names of the authors are also ambiguous as two authors with the same name does not necessary mean they are one and the same person. Second, there should be author-gene pairs which occur in just one publication. In these cases the inverse co-author graph could not help and contextual information has to be taken into account.

When we analysed the misclassified entities we found that most of the errors of two co-author graph-based methods could be eliminated by a sophisticated synonym matching algorithms. Our simple string matching approach, it transpires, has two main shortcomings. It does not handle the spelling variants of the gene aliases (an excellent work handling this task is [162]) and it does not deal with embedded named entities i.e. it matches gene names that are just a substring of a longer name like the name of a protein. The errors of the supervised systems (both the similarity-based and the decision tree-based ones) could probably be eliminated if bigger training sets were available.

Based on the promising results obtained so far from our study, we suppose that for abstracts the co-authorship information, the circumstances of the article's release (the journal, the year of publication) and a graph constructed above, can all be crucial building blocks for a sophisticated similarity measure among biological articles and therefore the methods introduced here ought to be useful for other biomedical natural language processing tasks as well. For example, we can reasonably assume that a biologist or biologist author group usually deals with the same special species. Hence a co-author graph-based method could be a powerful tool in the identification of the

organism dealing with in an article. In addition, all text classification and clustering tasks can achieve better results with a sophisticated similarity measure. Besides the biological Named Entity disambiguation tasks (which is also a document classification task), a task could for instance be one for target disease identification or protocol detection.

8.2 Response graphs in Opinion Mining

Opinion Mining [168] seeks to extract opinions and polarity about a certain topic from unstructured texts. This task has been attracting increasing academic interest in Natural Language Processing for over a decade. This phenomenon is due to the fact that people nowadays are more likely to share their emotions and opinions toward various topics, thus the amount of material on web sites (i.e. blogs and forums) that reflects the user's opinion has seen a remarkable growth in size [169]. From these rich sources of opinions valuable information can be extracted, which can help, for instance, political parties to design their campaign programme or companies to get feedback about their product structure-based on opinions expressed on the internet. Here, the goal is to capture not the objective content but the subjective sentiments, opinions and their polarity expressed in the text.

We developed a system [12] which classifies each member of a forum topic discussing the necessity and judgements about the Hungarian referendum on dual citizenship[1]. By reliably classifying the forum members, our aim was to predict the opinions of unknown people, thus to forecast the outcome of a forthcoming election.

We constructed the first opinion mining corpus for Hungarian for this task. The data for further processing was gathered from the posts of the forum topic of the Hungarian government portal (www.magyarorszag.hu) dealing with the referendum. We downloaded all the 1294 forum posts from the three month period preceding the referendum and these were annotated by two independent linguists. The annotators determined three categories of comments, i.e. *irrelevant*, *supporting* and *rejecting* ones. However, preliminary results revealed that a significant proportion of the posts belonged to another class, namely those stating that they would intentionally vote invalidly because they did not like the idea of asking such a question in a referendum. So, finally we had to classify the posts into four groups (*irrelevant*, *supporting*, *rejecting* and *invalid*).

We combined the results of two Machine Learning methods in our system. One of them was based on the traditional Vector Space Model (VSM), while the other one was trained on data derived from a so-called interaction or *response graph*.

[1] http://www.kettosallampolgarsag.mtaki.hu/english.html

Here our hypothesis was that people representing different views in the debate would comment more frequently on each other's posts compared to others. Thus, we composed a weighted, directed graph, in which each node is mapped to a person and the weight of an edge(A, B) corresponds to the number of person B's replies towards person A. We obtained this information from the HTML structure of the pages, but it is worth mentioning that not everyone indicated whether they were replying to another post and some people did not use this feature correctly (e.g. addressed replies to themselves, but such loops in the graph were omitted).

We extracted various characteristics from the graph that were used for creating feature vectors. These features consisted of the number of total/rejecting/supporting neighbours, the number of incoming and outcoming edges as well as the ratio of rejecting and supporting vertices one and two distance away.

Since the text-based (Vector Space Model) and the graph-based methods have different advantages and different disadvantages, it seemed obvious to combine their results. Predictions of the Vector Space Model-based Machine Learning model tend to achieve better results on the *irrelevant* class, but it performs quite well at other class labels as well. The results of the response graph tells us that it is better at recognising relevant forum members (belonging to *support*, *reject*, *invalid* classes) in those cases where users have more posts than the average number of posts per forum member (15.22 in this case). Our combination schema was simple: if a forum user had more than 15 posts, we accepted the prediction of the interaction graph, otherwise we chose the Vector Space Model-based prediction, improving the accuracy of our system in this way.

This hybrid system was able to achieve an overall accuracy of 71.76% in a one-member-leave-out evaluation scheme (the VSM-based and the graph-based systems alone achieved 65.88% and 55.29%, respectively). This means that we managed to outperform our baseline value of 34% by more than 37%, and we also successfully approximated the inter-annotator accuracy of 72.94%. It is also interesting to note that the ratio of the supporting forum members among the most relevant classes (*supporting* + *rejecting*) in the etalon dataset was 0.57 and the very same ratio resulted in 0.63 based on our predictions. Thus the results of our system might be used for giving a rough approximation on the outcome of such a forum-based debate.

8.3 Related work

In general, we did not find any published work on exploiting external graphical information (which cannot be derived from the text itself) for solving Natural Language Processing tasks.

There are several earlier studies available on general biomedical disambiguation tasks

like [170, 171, 172], to name but a few. Weeber et al. [170] annotated manually a UMLS-WSD corpus for supervised learning purposes. Savova et al. [171] introduced the utility of unlabeled data in general biomedical entity disambiguation. Their unsupervised approach looked for clusters among MedLine abstracts containing the target word, based on single word and bigram, first- and second order co-occurrence information. Liu et al. [172] built a train set automatically for each target term based on the co-occurrences of unambiguous synonyms in other documents.

When handling the particular GSD task, the AZuRE system [173] automatically assigns gene names to their LocusLink IDs based on the Naive Bayes model and contextual similarity. It extracted the training sets automatically from MedLine references in the LocusLink and SwissProt databases. Schijvenaars et al. [167] also generated the training set automatically from several existing databases. They built up their vector space from MeSH terms and gene names identified by string-matching then a cosine similarity metric based disambiguation was applied. The ProMiner [166] GN system contains a disambiguation module as well. It utilises the synonyms of the target gene name which are present in the document of the test gene.

Our GSD results are directly comparable just to Xu et al.'s results (we used the same datasets). They applied a vector space model with cosine similarity measure between the abstracts in question and the gene profiles which were in fact the centroids of the training instances. As they pointed out, there was not any significant information gain using the texts themselves along with the manually added MeSH codes. Table 8.4 lists the situation where only the features embedded in MedLine were used by both systems, but we achieved better results (with an average precision of 9.5% together with an average improvement of 1.5% in recall) than the best system of Xu et al. [34, 35], who employed external automatic annotation tools (MetaMap and BioMedLee) as well.

Studies concerned with the topic of Opinion Mining have only become of real academic interest in the past few years and no previous study was carried out on Opinion Mining for Hungarian. Our target application has some similarities with that described in [174] where the aim there was to predict the results of the forthcoming Canadian elections by collecting predictive opinions and deriving generalised features from them. We also worked on the election domain but in Hungarian, however, the main difference between the two studies is that we were interested in personal, subjective opinions towards the topic (e.g. "I strongly reject this issue and I will definitely say no at the referendum.") instead of predictive opinions (e.g. "I think the Democrats will win."). Other studies such as [175] focus on customer opinion extraction. Their task consisted of extracting aspect-evaluation and aspect-of relations from unstructured weblog posts on product reviews. They used contextual and context-independent clues as well. However, to the best of our knowledge, no other system to date combines textual information and non-textual inter-document relations in Opinion Mining.

8.4 Summary of thesis results

The exploitation potential of non-textual relations among documents for IE tasks were highlighted in this chapter.

The author examined the utility of the inverse co-author graph and experimentally demonstrated the utility of co-authorship analysis for the GSD task [11]. His hypothesis was that a biologist refers to exactly one gene by a fixed gene alias, and in experiments we found evidence for this. Moreover, he found that a disambiguation decision can be made in 85% of the cases with an extremely high precision rate (99.5%) by just using information obtained from the inverse co-author graph. If we need to build a GSD system with a full coverage, the co-authorship information can be incorporated into the system and by doing so eliminate about the half of the errors of the original system.

The author with his colleagues developed an Opinion Mining system which uses the information gathered from texts along with the response graph [12]. For this task the first corpus dedicated to Opinion Mining in Hungarian was constructed and results close to the inter-annotator agreement rate were achieved.

All the contributions described in [11] are the results of the author alone. In [12], the author's own contribution is the idea and general concept of using the response graph for Opinion Mining.

Chapter 9

Summary

9.1 Summary in English

The chief aim of this thesis was to examine various Machine Learning methods and discuss their suitability in real-world Information Extraction tasks. Among the Machine Learning tools, several less frequently used ones and novel ideas were experimentally investigated and discussed. The tasks themselves cover a wide range of different tasks from language-independent and multi-domain Named Entity recognition (word sequence labelling) to Name Normalisation and Opinion Mining.

The summary below, like the thesis itself, consists of two main parts. The first part summarises our findings in Supervised learning techniques for Information Extraction tasks and in the second part we describe work done in Exploitation strategies of external resources for Information Extraction tasks.

9.1.1 Supervised learning for Information Extraction tasks

When attempting to effectively solve classification problems it is worth applying various types of classification methods and strategies. We described comparative experiments on two main IE approaches (token-level and sequential models) using several learning algorithms on the *Named Entity Recognition* and *Metonymy Resolution* tasks.

The combination of individual learning models often leads to a 'better' model than those serving as a basis for it. We presented several experimental results on Named Entity Recognition tasks obtained via several meta-learning schemes and presented a novel scheme which is based on the split and recombination of the feature set. Based on the experiments with individual learners and meta-learning schemes, we constructed a complex statistical NER system which achieved state-of-the-art results on several datasets.

Supervised systems usually predicate well on unseen instances if they share the characteristics of the training dataset. Hence, when the target texts are changing, new

training datasets are required. We discussed several situations where the datasets are changing for a particular task, but the same learning procedure could be applied with minor modifications.

Results

Together with his colleagues, the author constructed a Machine Learning-based NER system [1, 4]. Our classification systems achieved results on international reference datasets which are competitive with other state-of-the-art NER taggers. The major contribution of the author here is an experimental comparison of the token-based versus sequential approaches, and several classification algorithms. On the whole, the author recommends using decision trees in the development phase (experiments) of an application because of its training time, ease of interpretability and use of generative models (Logistic Regression in classification and CRF in sequence labeling tasks) in the final versions.

Based on meta-learning experiments, the author cosiders that even simple combination schemes are worth employing because they usually give a significant accuracy improvement. The scores are usually better than those achieved by employing more sophisticated but time-consuming stand-alone learning algorithms. He introduced a novel combination approach called *Feature set split and recombination* [4], which played a key role in the construction of the complex NER system by the author with his colleagues. This NER system is competitive with the published state-of-the-art systems. It has a different theoretical background compared to the widely used sequential ones, which makes it an excellent candidate for a combination scheme with outer NER systems.

Later on, the author carried out several experiments where a training dataset was presented for a new domain but the same task. It was demonstrated that the Machine Learning model with minor domain-specific modifications can be applied successfully. Together with his colleagues, the author participated in the 2006 I2B2 shared task challenge on medical record de-identification [5] with this domain-adapted system and achieved top results.

9.1.2 Exploitation of external knowledge in Information Extraction tasks

Supervised methods requires a training corpus – with an appropriate size – for every task where the difference among tasks can be just marginal. In Part II, we investigate several approaches which seek to exploit knowledge from outside the training data, thus helping to decrease the required amount of training data to a minimum level.

The most widely used external resources in an Machine Learning task are unlabeled instances. The way of training model by using unlabeled data, together with labeled

data is called *semi-supervised learning*. Here, the goal is to utilize the unlabeled data during the training on labeled ones. We introduced our NER refinement and lemmatisation approaches which exploit the largest unlabeled corpus of the world, the WWW.

Besides unlabeled texts, existing expert decision systems, manually built taxonomies or written descriptions may contain useful information about the given Information Extraction task. An excellent example for this is clinical IE, where the knowledge of thousands years has been gathered into medical lexicons. We presented several ways of integrating the medical lexical knowledge into Machine Learning models – which are trained on free-text corpora – through two clinical IE applications, namely *ICD coding* and *obesity detection*.

In an applied Information Extraction task the documents to be processed are usually not independent of each other. The relations among documents can be exploited in the IE task itself. We discussed two tasks where graphs were constructed and employed based on these relations. In the biological *Gene Symbol Disambiguation* task we utilise the co-authorship graph, while in the *Opinion Mining* task the response graph will be built.

Results

The author with his colleagues developed WWW-based NER post-processing heuristics and experimentally investigated their use on general reference NE corpora[7]. They constructed several corpora for the English and Hungarian NE lemmatisation and separation tasks [8]. Based on these constructed corpora, automatically derived simple decision rules were introduced. Subsequent experiments confirmed that the result frequencies of search engines provide enough information to support such NE related tasks.

Later on, the author with his co-authors developed solutions for clinical Information Extraction tasks [9, 10], which integrate Machine Learning approaches and external knowledge sources. They exploit the advantages of expert systems and are able to handle rare labels effectively. 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, rarely used medical terms or other linguistic structures. Each statistical system along with the described integration methods developed for the two tasks are the author's own contributions.

The author examined the utility of the inverse co-author graph and experimentally demonstrated the utility of co-authorship analysis for the GSD task [11]. He found that a disambiguation decision can be made in 85% of the cases with an extremely high precision rate (99.5%) by just using information obtained from the inverse co-author graph. Later on, the author with his colleagues developed an Opinion Mining system which makes use of the information gathered from texts along with the response graph [12].

9.1.3 Conclusions of the Thesis

The key conclusions of this thesis are the following:

- The task-specific selection of Machine Learning methods is important. The huge amount of discrete features in Information Extraction tasks imply the use of decision trees or generative models.
- The trade-off between training time and accuracy is worth taking into account in the development period. Learners which train 10 times slower than simpler models achieve a relative 3-4% improvement in performance.
- Even simple (and fast) learner combination schemes can achieve significant improvements in performance.
- In the middle application layer of IE systems like NER – where the deep language-specific facts (like morphological and POS codes) are encoded into features and training sets are available – the statistical systems work language independently, while changing the domain in a certain language requires much more effort to achieve a satisfactory level of performance.
- A small change in the domain generally requires new manually labeled training corpus. Hence automatic adaptation techniques and/or approaches are required which reduce the training sample need by magnitudes.
- The WWW can be exploited as an external information (common knowledge) source in various Information Extraction tasks.
- Domain experts are mainly employed for providing labeled datasets in a supervised Machine Learning setting. We think that the the future of Information Extraction includes a more active involvement of these experts into the model-building process (interactive learning). We demonstrated this fact by integrating existing expert decision systems into data-driven models.
- There are several information sources available which contains important information for IE applications outside the text of the documents. As an example, we showed that graphical inter-document relations can significantly improve accuracy. We are of the view that the exploitation of this issue will be an emerging field of Information Extraction in the future.

9.2 Summary in Hungarian

Bevezetés

A disszertációban bemutattunk számos gépi tanulási technikát, és azok valós életbeli Információ-kinyerési problémákban való alkalmazhatóságát vizsgáltuk. A gépi tanulási módszerek közt ritkán alkalmazott eljárásokkal és újszerű technikákkal is kísérleteztünk. A tárgyalt feladatok széles skálát ölelnek fel, a nyelv-független névelem (Named Entity) felismeréstől (token-sorozatok címkézése) kezdve, a névelem normalizáción át a vélemény-detekcióig.

Az összefoglaló felépítése

Az összefoglaló szerkezete a tézis felépítését követi, a disszertáció két fő témáját tárgyalja. Az első rész (3-5 fejezetek) a felügyelt gépi tanulási módszereket ismerteti, míg a második (6-8 fejezetek) a tanító adatbázison kívüli információk felhasználására, rendszerbe integrálási lehetőségeire mutat példát.

Felügyelt gépi tanulási módszerek az Információ-kinyerésben

Minden gépi tanuló algoritmushoz adható olyan tanulási feladat, amelyiken más algoritmusok hatékonyabban teljesítenek [67], ezért érdemes a tanuló algoritmust feladatspecifikusan megválasztani. A dolgozatban bemutatott Információ-kinyerési problémák megoldása során alkalmazott két (token-alapú és szekvencia-alapú) megközelítési módot empirikusan összehasonlítottuk és számos osztályozó algoritmus hatékonyságát teszteltük többségében *névelem felismerés*i (Named Entity Recognition, NER) [1, 4] adathalmazokon. Ezen osztályozási problémáknak speciális tulajdonságai a nagy dimenziós (általában több tízezres) jellemzőtér, illetve a ritka és diszkrét jellemzők.

A *meta-tanuló*k különböző tanulók (tanuló algoritmusok példányai vagy ugyanazon algoritmus paraméterezett változatai) együttes alkalmazásával jönnek létre. Több tanuló-kombinációs módszer ismert, melyek általában jobb modellt eredményeznek, mint az alapjául szolgáló algoritmusok. Az ismert meta-tanulók alkalmazásával nyert eredmények mellett bemutatásra került egy újszerű eljárás is [4]. Ez a megközelítésünk néhány kisebb, átfedő jellemzőhalmazt választ ki az eredetei jellemzőtérből, majd az ezekkel tanított modelleket ötvözi.

A felügyelt gépi tanulási módszerek általában jó pontosságot érnek el az automatikus címkézési feladatokon ismeretlen szövegek esetén, ha a tesztszöveg karakterisztikája megegyezik a tanító adatbáziséval. Azonban, ha a célszövegek jellemzői megváltoznak (például gazdasági hírekről orvosi zárójelentésekre térünk át), akkor új tanító adatbázisra van szükség. A dolgozat 5. fejeztében több, nyelvben vagy a szövegek

témájában eltérő feladatot ismertettünk, amelyeknél a komplex NER rendszerünk finom módosításokkal igen jó eredményeket ért el.

Eredmények

A szerző és társai megterveztek és kifejlesztettek egy gépi tanulási módszereken alapuló *névelem-felismerő keretrendszer*t, amely több nemzetközi referencia adatbázison is kiemelkedő eredményt ért el [1, 4]. A rendszer tervezésének egyik alappillére volt gépi tanulási algoritmusok viselkedésének empirikus vizsgálata, amit a szerző különböző Információ-kinyerési problémákon végzett el. Összességében a szerző döntési fák alkalmazását javasolja a fejlesztés, kísérletezés folyamán, annak gyors tanítási ideje és a tanult modell interpretálhatósága miatt. A generatív modellek (pl.: Logisztikus Regresszió [50], Feltételes Valószínűségi Mezők [52]) általában csak néhány százalékkal teljesítenek jobban (hosszabb tanítási idő árán), így azok végső modellként való alkalmazása ajánlott.

A szerző empirikus vizsgálatokkal bizonyította, hogy egyszerű *tanuló-kombinációs sémá*k is szignifikáns javulást eredményeznek. Ezen kombinációs sémák egyszerű, gyors alap-tanulókat használva is általában jobb eredményeket képesek elérni, mint a szofisztikáltabb, időigényesebb tanulók önmagukban. A kidolgozott, komplex NER rendszer is ilyen meta-tanuló algoritmusok alkalmazására épül. A 4. fejezetben bemutatásra került a szerző újszerű kombinációs algoritmusa, amely a jellemzőtér felosztásán és a tanult modellek kombinációján alapul [4]. A komplex rendszer több adatbázison versenyképes eredményt ért el a legjobb, publikált rendszerekhez hasonlítva, míg azoktól különböző elmélti alapokon nyugszik. Ez utóbbi tény különösen alkalmassá teszi más külső NER rendszerekkel való kombinálásra.

A szerző és társai a NER rendszert eredményesen alkalmazták orvosi zárójelentések szövegein is (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 [5].

Külső információs források kiaknázása Információ-kinyerési feladatokban

A felügyelt tanulásnál minden feladathoz szükség van egy megfelelő méretű tanító adatbázisra, még akkor is ha a feladatok csak kis mértékben térnek el egymástól (pl.: NER gazdasági és sporthíreken). A dolgozat II. részében különböző külső információs források felhasználási lehetőségeit vizsgáltuk Információ-kinyerési problémák megoldására.

Ezen kísérletek célja, hogy a szükséges tanító példák számát minimalizáljuk, így az újabb problémákra történő adaptáció gyorsabbá és költséghatékonyabbá válik.

A legtöbbet kutatott ilyen terület a *részben felügyelt tanulás*, ahol a jelölt tanító adatbázis mellett jelöletlen adatból is próbálunk hasznos információt kinyerni. Ehhez kapcsolódva azt vizsgáltuk, hogy az Internetet, mint a világ legnagyobb jelöletlen szöveges adatbázisát, hogyan lehet felhasználni névelem-felismerési problémákhoz [7] illetve tulajdonnév-lemmatizáláshoz [8].

Egy másik, igen hasznos (de kevésbé vizsgált külső információforrás) az emberi erőforrással épített taxonómiák, döntési szabályrendszerek, mint például a klinikai információkinyerés során alkalmazható, ezer évek tudását magába foglaló orvosi enciklopédiák. A dolgozatban bemutattunk több módszert, amelyekben ilyen szabályrendszereket használtunk fel gépi tanulási modellek támogatására az automatikus BNO (Betegségek Nemzetközi Osztályozása) kódolási [9] és betegség-azonosítási feladatokban [10].

Alkalmazott Információ-kinyerési problémáknál a feldolgozandó dokumentumok általában nem függetlenek egymástól. A dokumentumok közti, gráf jellegű, külső információk alkalmazását ismerteti a dolgozat utolsó fejezete. Ezen adatok felhasználásával szignifikáns javulás érhető el, amit génnév-egyértelműsítési – ahol a gráf itt a társszerzőséget reprezentálja – [11] és vélemény-detekciós feladaton – ahol a gráf fórumozók egymásra reagálását fejezi ki – [12] szemléltettünk.

Eredmények

A szerző társaival több Internet-gyakoriság-alapú NER utófeldolgozó algoritmust dolgozott ki, aminek hatékonyságát empirikusan validálta referencia adatbázisokon [7]. Hasonló statisztikai módszer segítségével egy tulajdonnév-lemmatizálási eljárás is kidolgozásra került [8], melynek eredményei igazolják, hogy habár a WWW igen zajos, a redundanciát kihasználva hasznos információval szolgálhat különböző Információ-kinyerő feladatokon.

Ezt követően a szerző és társa egy kórházi leletek betegségkódokkal, illetve BNO-kódokkal való automatikus címkézésére alkalmas rendszert fejlesztett ki. Ez a rendszer egy automatikus klinikai kódoló rendszerek kiértékelésére szervezett versenyen a legjobb pontosságot érte el [16]. A verseny tapasztalatai alapján a szerző és társa egy szakértői és statisztikai rendszerek kombinációján alapuló modellt dolgozott ki, mely képes a rendelkezésre álló szabályalapú rendszereket címkézett példák felhasználásával tovább pontosítani, fejleszteni [9, 10]. Az ide kapcsolódó kísérletek során a szerző hozzájárulása a gépi tanulási modellek kiválasztásában és implementálásában, a szabályalapú rendszerek integrálásában és kivitelezésében volt meghatározó.

A szerző az ún. szerzőségi gráf felhasználásával igen jó eredményeket ért el a génnév-egyértelműsítési feladaton (a gráf alapján az esetek 85%-ában 99,5%-os pontossággal hozható meg az egyértelműsítési döntés) [11]. Hasonló módszer segítségével, a szerző

társaival egy vélemény-detekciós probléma megoldása során a fórumozók válaszadási-gráfjából nyert ki információt [12].

Konklúzió

A disszertáció főbb konklúziói a következő pontokba foglalhatók össze:

- A gépi tanuló algoritmusok feladat-specifikus kiválasztása nagyon fontos. Az Információ-kinyerési feladatoknál a jellemzőtér nagy mennyiségű diszkrét jellemzőt tartalmaz, ami döntési fák és generatív modellek használatát implikálja.
- A rendszerek fejlesztési fázisában érdemes gyors tanulókat használni, hiszen a tízszer lassabban tanuló szofisztikáltabb módszerek csak 3-4%-al jobb végső pontosság elérésére képesek.
- Egyszerű (és gyors) meta-tanulók is szignifikáns javulást hozhatnak a tanulási folyamatba.
- Az Információ-kinyerés középső rétegeiben – ahol a nyelv-specifikus információk (morfológiai jegyek, POS kódok) már a jellemzőtérbe vannak kódolva – ugyanazon statisztikai rendszer tulajdonképpen változtatás nélkül több nyelvre is ugyanolyan pontossággal működik. A szöveg domainjének megváltozása esetén a rendszeren nagyobb adaptációs lépéseket kell elvégezni.
- A domain kis változása is új tanító adatbázist követelhet meg, ezért automatikus adaptációs technikákra és/vagy olyan algoritmusokra van szükség, amelyekkel a tanító példák számának szükségletét nagyságrendekkel csökkenthető.
- A WWW, mint külső információforrás (általános tudásbázis) hatékonyan kiaknázható számos Információ-kinyerési feladatban.
- A humán szakértők általában csak offline módon tanító példákat adnak a gépi tanuló algoritmusok számára. A szerző úgy véli, hogy a szakértők aktívabb bevonása a modellépítési folyamatba (interaktív tanulás) egy fontos jövőbeli kutatási irány lesz az Információ-kinyerés területén (ennek hatékonyságát volt hivatott demonstrálni a szabályalapú és statisztikai módszerek integrálása a klinikai területen).
- Számos adatforrás létezik a tanító adatbázison kívül, ami hasznos információt hordozhat az Információ-kinyerési alkalmazások számára. Példaként megmutattuk, hogy dokumentumközi információk felhasználásával a szöveg-alapú rendszerek szignifikánsan javíthatók.

Bibliography

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

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

[3] 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].

[4] Szarvas Gy, Farkas R, Kocsor A: **A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms**. *DS2006, LNAI* 2006, **4265**:267–278.

[5] Szarvas Gy, Farkas R, Busa-Fekete R: **State-of-the-art anonymisation of medical records using an iterative machine learning framework**. *Journal of the American Medical Informatics Association* 2007, **14**(5):574–580, [http://www.jamia.org/cgi/content/abstract/M2441v1].

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

[7] Farkas R, Szarvas Gy, Ormándi R: **Improving a State-of-the-Art Named Entity Recognition System Using the World Wide Web**. *ICDM2007, LNCS* 2007, **4597**:163–172.

[8] Farkas R, Vincze V, Nagy I, Ormándi R, Szarvas Gy, Almási A: **Web based lemmatisation of Named Entities.** In *Proceedings of the 11th International Conference on Text, Speech and Dialogue* 2008:53–60.

[9] 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].

[10] Farkas R, Szarvas Gy, Hegedűs I, Almási A, Vincze V, Ormándi R, Busa-Fekete R: **Semi-automated construction of decision rules to predict morbidities from clinical texts.** *Journal of the American Medical Informatics Association* 2009, **accepted for publication**.

[11] Farkas R: **The strength of co-authorship in gene name disambiguation**. *BMC Bioinformatics* 2008, **9**, [http://dx.doi.org/10.1186/1471-2105-9-69].

[12] Berend G, Farkas R: **Opinion Mining in Hungarian based on textual and graphical clues.** In *Proceedings of the 4th International Symposium on Data Mining and Intelligent Informaion Processing* 2008.

[13] Han J, Kamber M: *Data Mining. Concepts and Techniques.* Morgan Kaufmann, 2nd ed. edition 2006.

[14] Chinchor NA: **MUC-7 Named Entity Task Definition**. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* 1998. [Http://www.itl.nist.gov/iaui/894.02/related_projects/muc/].

[15] Hirschman L, Yeh A, Blaschke C, Valencia A: **Overview of BioCreAtIvE: critical assessment of information extraction for biology.** *BMC Bioinformatics* 2005, **6 Suppl 1**.

[16] 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].

[17] Program ACE: . [Http://www.itl.nist.gov/iad/894.01/tests/ace/].

[18] Gábor P: **NewsPro: automatikus információszerzés gazdasági rövidhírekből** . In *I. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'03)*. Edited by Alexin Z, Csendes D, Szeged, Hungary: SZTE, Informatikai Tanszékcsoport 2003:161–167.

[19] Sekine S, Sudo K, Nobata C: **Extended Named Entity Hierarchy** 2002, [citeseer.ist.psu.edu/sekine02extended.html].

[20] Hastie T, Tibshirani R, Friedman J: *The Elements of Statistical learning.* Springer 2003. [Chapter 7].

[21] Vapnik VN: *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc. 1995.

[22] Sang TK, F E, 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.

[23] Daelemans W, Zavrel J, van der Sloot K, van den Bosch A: **MBT: Memory-Based Tagger, version 1.0, Reference Guide.** Technical Report ILK-0209, University of Tilburg, The Netherlands 2002.

[24] Csendes D, Csirik J, Gyimóthy T, Kocsor A: **The Szeged Treebank**. In *Proceedings of the 8th International Conference on Text, Speech and Dialogue* 2005:123–131.

[25] Kuba A, Hócza A, Csirik J: **POS Tagging of Hungarian with Combined Statistical and Rule-Based Methods**. In *Proceedings of the 7th International Conference on Text, Speech and Dialogue* 2004:113–120.

[26] Uzuner O, Luo Y, Szolovits P: **Evaluating the State-of-the-Art in Automatic De-identification**. *Journal of the American Medical Informatics Association* 2007, **14**(5):550–563, [http://www.jamia.org/cgi/content/abstract/14/5/550].

[27] Markert K, Nissim M, Lw BPE: **Metonymy resolution as a classification task.** In *Proceedings of EMNLP* 2002:204–213.

[28] 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].

[29] Burnard L: *Users' Reference Guide, British National Corpus*. BNC Consortium, Oxford, England 1995.

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

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

[32] Hirschman L, Colosimo M, Morgan A, Yeh A: **Overview of BioCreAtIvE task 1B: normalized gene lists**. *BMC Bioinformatics* 2005, **6**(Suppl 1):S11.

[33] Xu H, Markatou M, Dimova R, Liu H, Friedman C: **Machine learning and word sense disambiguation in the biomedical domain: design and evaluation issues**. *BMC Bioinformatics* 2006, **7**:334, [http://www.biomedcentral.com/1471-2105/7/334].

[34] Xu H, Fan JW, Hripcsak G, Mendonca EA, Markatou M, Friedman C: **Gene symbol disambiguation using knowledge-based profiles.** *Bioinformatics* 2007, **23**(8):1015–1022, [http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=17314123].

[35] Xu H, Fan JW, Friedman C: **Combining multiple evidence for gene symbol disambiguation**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:41–48, [http://www.aclweb.org/anthology/W/W07/W07-1006].

[36] Maglott DR, Ostell J, Pruitt KD, Tatusova TA: **Entrez Gene: gene-centered information at NCBI.** *Nucleic Acids Research* 2007, **35**(Database-Issue):26–31, [http://dblp.uni-trier.de/db/journals/nar/nar35.html#MaglottOPT07].

[37] Morgan A, Wellner B, Colombe J, Arens R, Colosimo M, Hirschman L: **Evaluating the automatic mapping of human gene and protein mentions to unique identifiers.** *Pac Symp Biocomput* 2007.

[38] Quinlan JR: *C4.5: Programs for machine learning*. Morgan Kaufmann 1993.

[39] Quinlan JR: **Induction of decision trees.** *Machine Learning* 1986, **1**:81–106.

[40] Witten IH, Frank E: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann 1999, [http://www.amazon.de/exec/obidos/ASIN/1558605525].

[41] Bishop CM: *Neural Networks for Pattern Recognition*. Oxford University Press, Inc. 1995.

[42] Vapnik VN: *Statistical Learning Theory*. John-Wiley & Sons Inc. 1998.

[43] 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.

[44] Mercer J: *Functions of positive and negative type and their connection with the theory of integral equations*. Philos. Trans. Roy. Soc. London 1909.

[45] Curran JR, Clark S: **Language Independent NER using a Maximum Entropy Tagger**. In *Proceedings of CoNLL-2003*. Edited by Daelemans W, Osborne M, Edmonton, Canada 2003:164–167.

[46] Halácsy P, Kornai A, Németh L, Rung A, Szakadát I, Trón V: **A szószablya projekt – www.szoszablya.hu**. In *Proceedings of I. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)* 2003:298–299.

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

[48] 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. [Held in conjunction with COLING'2004].

[49] Alpaydin E: *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press 2004.

[50] le Cessie S, van Houwelingen J: **Ridge Estimators in Logistic Regression.** *Applied Statistics* 1992, **41**:191–201.

[51] John GH, Langley P: **Estimating continuous distributions in bayesian classifiers.** In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann 1995:338–345.

[52] Sutton C, Mccallum A: **Introduction to Conditional Random Fields for Relational Learning.** In *Introduction to Statistical Relational Learning*. Edited by Getoor L, Taskar B, MIT Press 2006.

[53] Lafferty J, McCallum A, Pereira F: **Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data**. In *Proc. 18th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA 2001:282–289, [citeseer.ist.psu.edu/lafferty01conditional.html].

[54] Farkas R, Szarvas Gy, Csirik J: **Special Semi-Supervised Techniques for Natural Language Processing Tasks.** In *Proceedings of the 6th International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics* 2007:360–365.

[55] Borthwick A, Sterling J, Agichtein E, Grishman R: **Description of the MENE Named Entity System as Used in MUC-7**. In *Proceedings of the Seventh Message Understanding Conference* 1998.

[56] 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.

[57] C Lee WJH, Chen HH: **Annotating Multiple Types of Biomedical Entities: A Single Word Classification Approach**. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)* 2004.

[58] Sekine S: **NYU: Description of the Japanese NE system used for MET-2**. In *Proc. of the Seventh Message Understanding Conference (MUC-7* 1998.

[59] Uchimoto K, Ma Q, Murata M, Ozaku H, Isahara H: **Named entity extraction based on a maximum entropy model and transformation rules**. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics 2000:326–335.

[60] Asahara M, Matsumoto Y: **Japanese Named Entity Extraction with Redundant Morphological Analysis**. In *Proceedings of Human Language Technology Conference(HLT-NAACL* 2003:8–15.

[61] Bikel DM, Miller S, Schwartz R, Weischedel R: **Nymble: a high-performance learning name-finder**. In *Proceedings of the Fifth Conference on Applied Natural Language Processing* 1997:194–201.

[62] Manning CD, Schütze H: *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press 1999, [citeseer.ist.psu.edu/635422.html].

[63] McCallum A, Freitag D, Pereira FCN: **Maximum Entropy Markov Models for Information Extraction and Segmentation**. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 2000:591–598.

[64] Mccallum A: **Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons**. In *Proceedings of CoNLL* 2003.

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

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

[67] Wolpert DH, Waters R: **The relationship between PAC, the statistical physics framework, the Bayesian framework, and the VC framework**. In *In*, Addison-Wesley 1994:117–214.

[68] Breiman L: **Bagging predictors.** In *Machine Learning* 1996:123–140.

[69] Wolpert DH: **Stacked generalization**. *Neural Networks* 1992, **5**:241–259.

[70] Schapire RE: **The Strength of Weak Learnability**. In *Machine Learning*, *Volume 5* 1990:197–227, [citeseer.ist.psu.edu/schapire90strength.html].

[71] Freud Y, Schapire R: **Experiments with a new boosting algorithm.** In *Machine Learning: Proceedings of the thirteenth international conference* 1996:148–156.

[72] Chieu HL, Ng HT: **Named entity recognition with a maximum entropy approach**. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003* 2003:160–163.

[73] 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.

[74] Mitchell T: **The Role of Unlabeled Data in Supervised Learning**. In *Proceedings of the Sixth International Colloquium on Cognitive Science* 1999. [(invited paper)].

[75] Carreras X, Màrques L, Padró L: **Named Entity Extraction using AdaBoost**. In *Proceedings of CoNLL-2002*, Taipei, Taiwan 2002:167–170.

[76] Lazarevic A: **Feature Bagging for Outlier Detection**. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* 2005:157–166.

[77] Sutton C, Sindelar M, McCallum A: **Feature Bagging: Preventing Weight Undertraining in Structured Discriminative Learning**. Technical Report IR 402, University of Massachusetts 2005.

[78] Ji H, Grishman R: **Data Selection in Semi-supervised Learning for Name Tagging**. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, Sydney, Australia: Association for Computational Linguistics 2006:48–55, [http://www.aclweb.org/anthology/W/W06/W06-0206].

[79] Zhang Z: **Weakly-supervised relation classification for information extraction**. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, New York, NY, USA: ACM 2004:581–588.

[80] Lin W, Yangarber R, Grishman R: **Bootstrapped Learning of Semantic Classes from Positive and Negative Examples**. In *Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data* 2003:103–111.

[81] Cucerzan S, Yarowsky D: **Language independent named entity recognition combining morphological and contextual evidence.** In *Proceedings of Joint SIGDAT Conference on EMNLP and VLC* 1999.

[82] 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.

[83] Huang F: **Multilingual Named Entity Extraction and Translation from Text and Speech**. *PhD thesis*, Carnegie Mellon University 2005.

[84] Wang L, Li W, Chang C: **Recognizing Unregistered Names for Mandarin Word Identification**. In *Proc. of COLING92*, COLING 1992:1239–1243.

[85] Petasis G, Vichot F, Wolinski F, Paliouras G, Karkaletsis V, Spyropoulos CD: **Using machine learning to maintain rule-based named-entity recognition and classification systems**. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics 2001:426–433.

[86] Popov B, Kirilov A, Maynard D, Manov D: **Creation of reusable components and language resources for Named Entity Recognition in Russian**. In *Proc. Conference on Language Resources and Evaluation* 2004.

[87] da Silva JF, Kozareva Z, Gabriel J, Lopes P: **Cluster Analysis and Classification of Named Entities**. In *Proc. Conference on Language Resources and Evaluation* 2004.

[88] Whitelaw C, Patrick J: **Evaluating Corpora for Named Entity Recognition Using Character-Level Features**. In *Australian Conference on Artificial Intelligence* 2003:910–921.

[89] Piskorski J: **Extraction of Polish Named-Entities**. In *Proceedings of International Conference on Language Resources an Evaluation - LREC 2004* 2004.

[90] Tóth K, Farkas R, Kocsor A: **Sentence Alignment of Hungarian-English Parallel Corpora Using a Hybrid Algorithm.** *Acta Cybernetica* 2008, **18**(3):463–478.

[91] Gale WA, Church KW: **A Program for Aligning Sentences in Bilingual Corpora**. In *Proceedings of the 11th Meeting of the Association for Computational Linguistics* 1991:177–184, [citeseer.ist.psu.edu/gale91program.html].

[92] Brown PF, Lai JC, Mercer RL: **Aligning Sentences in Parallel Corpora**. In *Proceedings of the 29th Meeting of the Association for Computational Linguistics* 1991:169–176, [citeseer.ist.psu.edu/brown91aligning.html].

[93] Chen SF: **Aligning Sentences in Bilingual Corpora Using Lexical Information**. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio 1993:9–16.

[94] Kay M, Röscheisen M: **Text-Translation Alignment**. In *Computational Linguistics, Volume 19* 1993:121–142.

[95] Simard M, Foster G, Isabelle P: **Using Cognates to Align Sentences in Bilingual Corpora**. In *Proceedings of the Fourth International Conference on Theoretical and Methodogical Issues in Machine translation (TMI92), (Montreal)* 1992:67–81, [citeseer.ist.psu.edu/simard92using.html].

[96] Pohl G: **Szövegszinkronizációs módszerek, hibrid bekezdés– és mondatszinkronizációs megoldás**. In *Proceedings of Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)* 2003:254–259.

[97] Maynard D, Tablan V, Cunningham H: **NE Recognition Without Training Data on a Language You Don't Speak**. In *Proceedings of Workshop on Multilingual and Mixed-language Named Entity Recognition, ACL 2003* 2003.

[98] Maynard D, Bontcheva K, Cunningham H: **Automatic Language-Independent Induction of Gazetteer Lists**. In *Proceedings of 4th Language Resources and Evaluation Conference* 2004.

[99] Buchholz S, Marsi E: **CoNLL-X Shared Task on Multilingual Dependency Parsing**. In *Proceedings CoNLL-X* 2006[http://www.cnts.ua.ac.be/conll/proceedings.html,http://www.cnts.ua.ac.be/conll/pdf/14964.pdf].

[100] Nivre J, Hall J, Kübler S, McDonald R, Nilsson J, Riedel S, Yuret D: **The CoNLL 2007 Shared Task on Dependency Parsing**. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, Czech Republic: Association for Computational Linguistics 2007:915–932, [http://www.aclweb.org/anthology/D/D07/D07-1096].

[101] Sweeney L: **Replacing Personally-Identifying Information in Medical Records, the Scrub System**. In *Cimino, J., ed. Proceedings, Journal of the American Medical Informatics Assoc*, Hanley & Belfus 1996:333–337.

[102] Ruch P, Baud RH, Rassinoux AM, Bouillon P, Robert G: **Medical Document Anonymization with a Semantic Lexicon**. In *Proceedings of the AMIA Annual Symposium* 2000:729–733.

[103] 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.

[104] Gupta D, Saul M, Gilbertson J: **Evaluation of a deidentification (De-Id) software engine to share pathology reports and clinical documents for research.** *American Journal of Clinical Pathology* 2004, **121(21)**(1-3):169–71, [citeseer.ist.psu.edu/bikel99algorithm.html].

[105] Sibanda T, Uzuner O, Uzuner O: **Role of Local Context in Automatic Deidentification of Ungrammatical, Fragmented Text**. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, New York City, USA: Association for Computational Linguistics 2006:65–73, [http://www.aclweb.org/anthology/N/N06/N06-1009].

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

[107] 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.

[108] 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. *Journal of the American Medical Informatics Association* 2007, **14**(5):564–573, [http://www.jamia.org/cgi/content/abstract/14/5/564].

[109] 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.

[110] 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.

[111] Varga D, Halacsy P, Kornai A, Nagy V, Nemeth L, Tron V: **Parallel corpora for medium density languages**. In *Proceedings of RANLP'2005*, Borovets, Bulgaria 2005:590 – 596.

[112] Chapman W, Bridewell W, Hanbury P, Cooper G, Buchanan B: **A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries**. *Journal of Biomedical Informatics* 2001, **34**(5):301–310.

[113] Mutalik P, Deshpande A, Nadkarni P: **Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study Using the UMLS**. *Journal of the American Medical Informatics Association* 2001, **8**(6):598–609.

[114] Friedman C, Alderson P, Austin J, Cimino J, Johnson S: **A general natural-language text processor for clinical radiology**. *Journal of the American Medical Informatics Association* 1994, **1**(2):161–174.

[115] Elkin P, Brown S, Bauer B, Husser C, Carruth W, Bergstrom L, Wahner-Roedler D: **A controlled trial of automated classification of negation from clinical notes**. *BMC Medical Informatics and Decision Making* 2005, **5**:13. [Doi:10.1186/1472-6947-5-13].

[116] Huang Y, Lowe H: **A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports**. *Journal of the American Medical Informatics Association* 2007, **14**(3):304–311.

[117] Hyland K: **Hedging in academic writing and EAP textbooks**. *English for Specific Purposes* 1994, **13**(3):239–256.

[118] Light M, Qui X, Srinivasan P: **The language of bioscience: Facts, speculations, and statements in between**. *Proceedings of BioLink 2004 Workshop on*

Linking Biological Literature, Ontologies and Databases: Tools for Users 2004, :17–24.

[119] Medlock B, Briscoe T: **Weakly supervised learning for hedge classification in scientific literature**. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* 2007, :992–999.

[120] Kim J, Ohta T, Tsujii J: **Corpus annotation for mining biomedical events from literature**. *BMC Bioinformatics* 2008, **9**:10.

[121] Pyysalo S, Ginter F, Heimonen J, Bjorne J, Boberg J, Jarvinen J, Salakoski T: **BioInfer: a corpus for information extraction in the biomedical domain**. *BMC Bioinformatics* 2007, **8**:50. [Doi:10.1186/1471-2105-8-50].

[122] Zweigenbaum P: **Natural Language Processing in the Medical and Biological Domains: a Parallel Perspective.** In *Third International Symposium on Semantic Mining in Biomedicine* 2008. [(invited talk)].

[123] Zhu X: **Semi-Supervised Learning Literature Survey**. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison 2005.

[124] Cozman FG, Cohen I, Cirelo MC: **Semi-Supervised Learning of Mixture Models**. In *ICML* 2003:99–106.

[125] Vapnik VN: *Statistical Learning Theory*. New York, NY, USA: Wiley 1998.

[126] Chapelle O, Zien A: **Semi-Supervised Classification by Low Density Separation**. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* 2005, :57–64, [http://www.gatsby.ucl.ac.uk/aistats/fullpapers/198.pdf].

[127] Linguistic Data Consortium (LDC): **The English Gigaword Corpus**. [Http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05].

[128] Etzioni O, Cafarella MJ, Downey D, Popescu A, Shaked T, Soderland S, Weld DS, Yates A: **Unsupervised named-entity extraction from the Web: An experimental study.** *Artificial Intelligence* 2005, **165**:91–134, [http://www.cs.washington.edu/homes/etzioni/papers/knowitall-aij.pdf].

[129] Shinzato K, Sekine S, Yoshinaga N, Torisawa K: **Constructing Dictionaries for Named Entity Recognition on Specific Domains from the Web.** In *Proceedings of ISWC'06 Workshop on Web Content Mining with Human Language Technologies* 2006.

[130] Hearst M: **Automatic Acquisition of Hyponyms from Large Text Corpora.** In *Proceedings of the 14th International Confernece on Computational Linguistics* 1992.

[131] Fellbaum C (Ed): *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press 1998, [http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/026206197X].

[132] Aha DW, Kibler D, Albert MK: **Instance-Based Learning Algorithms**. *Machine Learning* 1991, **6**:37–66.

[133] Melĉuk I: **Modèle de la déclinaison hongroise**. In *Cours de morphologie générale (théorique et descriptive). Vol. 5.*, Montréal, Les Presses de l'Université de Montréal, CNRS Éditions 2000:191–261.

[134] Halácsy P, Trón V: **Benefits of Resource-Based Stemming in Hungarian Information Retrieval**. In *CLEF* 2006:99–106.

[135] Cimiano P, Handschuh S: **Towards the Self-Annotating Web**. In *Proceedings of World Wide Web Conference 2004* 2004:462–471.

[136] Bunescu RC, Pasca M: **Using Encyclopedic Knowledge for Named entity Disambiguation**. In *EACL* 2006.

[137] Artiles J, Gonzalo J, Sekine S: **The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task**. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic: Association for Computational Linguistics 2007:64–69, [http://www.aclweb.org/anthology-new/W07/W07-2012.bib].

[138] 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].

[139] Erjavec T, Dzeroski S: **Machine Learning of Morphosyntactic Structure: Lemmatizing Unknown Slovene Words.** *Applied Artificial Intelligence* 2004, **18**:17–41, [http://dblp.uni-trier.de/db/journals/aai/aai18.html#ErjavecD04].

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

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

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

[143] 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].

[144] 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.

[145] Uzuner O: **Recognizing Obesity and Co-morbidities in Sparse Data.** *Journal of American Medical Informatics Association* 2009.

[146] Barness L, Opitz J, E GB: **Obesity: genetic, molecular, and environmental aspects.** *American Journal of Medical Genetics* 2007, **Part A, Volume 143A**(24):3016–3034.

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

[148] 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.

[149] de Lima LR, 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.

[150] Suominen H, Ginter F, Pyysalo S, Airola A, Pahikkala T, Salanterä S, Salakoski T: **Machine Learning to Automate the Assignment of Diagnosis Codes to Free-text Radiology Reports: a Method Description**. In *Proceedings of the ICML/UAI workshop on Machine Learning in health care applications* 2008.

[151] 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].

[152] 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].

[153] 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].

[154] Savova G, Clark C, Zheng J, Cohen K, Murphy S, Wellner B, Harris D, Lazo M, Aberdeen J, Hu Q, Chute C, Hirschman L: **The Mayo/MITRE System for Discovery of Obesity and Its Comorbidities.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[155] Childs L, Taylor R, Simonsen L, Heintzelman N, Kowalski K, Enelow R: **Description of the Lockheed Martin / SAGE Analytica System for the i2b2 Challenge in Natural Language Processing for Clinical Data.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[156] Solt I, Tikk D, Gál V, Kardkovács Z: **Context-Aware Rule Based Classifier for Semantic Classification of Diseases in Discharge Summaries.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[157] Yang H, Spasic I, Keane J, Nenadic G: **Combining Lexical Profiling, Rules and Machine Learning for Disease Prediction from Hospital Discharge Summaries.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[158] Peshkin L, Cano C, Carpenter B, Baldwin B: **Regularized Logistic Regression for Clinical Record Processing.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[159] Califf M: **Combining Rules and Naive Bayes for Disease Classification.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[160] Matthews M: **Bayesian Networks and the i2b2 Obesity Challenge.** In *Proceedings of the i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.* 2008.

[161] Yeh AS, Hirschman L, Morgan AA: **Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup**. *CoRR* 2003, **cs.CL/0308032**, [http://dblp.uni-trier.de/db/journals/corr/corr0308.html#cs-CL-0308032].

[162] Hakenberg J: **What's in a gene name? Automated refinement of gene name dictionaries**. In *Biological, translational, and clinical language processing*, Prague, Czech Republic: Association for Computational Linguistics 2007:153–160, [http://www.aclweb.org/anthology/W/W07/W07-1020].

[163] Chen L, Liu H, Friedman C: **Gene name ambiguity of eukaryotic nomenclatures.** *Bioinformatics* 2005, **21**(2):248–256, [http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics21.html#ChenLF05].

[164] Agirre E, Edmonds P (Eds): *Word Sense Disambiguation: Algorithms and Applications*, *Volume 33 of* Text, Speech and Language Technology. Springer 2006, [http://www.amazon.co.uk/exec/obidos/ASIN/1402048084/citeulike-21].

[165] Barabasi AL, Jeong H, Neda Z, Ravasz E, Schubert A, Vicsek T: **Evolution of the social network of scientific collaborations**. *Physica A: Statistical Mechanics and its Applications* 2002, **311**(3-4):590–614.

[166] Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J: **ProMiner: rule-based protein and gene entity recognition.** *BMC Bioinformatics* 2005, **6 Suppl 1**.

[167] Schijvenaars B, Mons B, Weeber M, Schuemie M, van Mulligen E, Wain H, Kors J: **Thesaurus-based disambiguation of gene symbols.** *BMC Bioinformatics* 2005, **6**.

[168] Ghose A, Ipeirotis PG, Sundararajan A: **Opinion Mining using Econometrics: A Case Study on Reputation Systems**. In *ACL* 2007.

[169] Metcalfe R: **Metcalfe's law: A network becomes more valuable as it reaches more users**. *Infoworld* 1995, **17**.

[170] Weeber M, Mork J, Aronson A: **Developing a test collection for biomedical word sense disambiguation.** *Proc AMIA Symp* 2001, :746–750.

[171] Savova G, Pedersen T, Purandare A, Kulkarni A: **Resolving Ambiguities in Biomedical Text with Unsupervised Clustering Approaches.** Research

Report UMSI 2005/80 and CB Number 2005/21, University of Minnesota Supercomputing Institute 2005.

[172] Liu H, Lussier YA, Friedman C: **Disambiguating Ambiguous Biomedical Terms in Biomedical Narrative Text: An Unsupervised Method.** *Journal of Biomedical Informatics* 2001, **34**(4):249–261, [http://dblp.uni-trier.de/db/journals/jbi/jbi34.html#LiuLF01].

[173] Podowski RM, Cleary JG, Goncharoff NT, Amoutzias G, Hayes WS: **AZuRE, a Scalable System for Automated Term Disambiguation of Gene and Protein Names**. In *CSB '04: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB'04)*, Washington, DC, USA: IEEE Computer Society 2004:415–424.

[174] Kim SM, Hovy E: **Crystal: Analyzing Predictive Opinions on the Web**. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic: Association for Computational Linguistics 2007:1056–1064, [http://www.aclweb.org/anthology/D/D07/D07-1113].

[175] Kobayashi N, Inui K, Matsumoto Y: **Extracting Aspect-Evaluation and Aspect-Of Relations in Opinion Mining**. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic: Association for Computational Linguistics 2007:1065–1074, [http://www.aclweb.org/anthology/D/D07/D07-1114].