LUDWIG-MAXIMILIANS-UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN

Institut für Informatik
Lehr- und Forschungseinheit für
Datenbanksysteme

Diplomarbeit
in Bioinformatik

Entity- and relation extraction from biomedical text corpora

Markus Bundschus

Aufgabensteller:	Prof. Dr. Hans-Peter Kriegel
Betreuer:	Dr. Volker Tresp, Dr. Kai Yu, Dr. Peer Kröger, Arthur Zimek
Abgabedatum:	15.10.2006

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

15.10.2006 _____

Markus Bundschus

Abstract

This thesis addresses the problem of named entity recognition and the problem of relation extraction from biomedical text corpora. Named entity recognition (NER) and relation extraction (RE) are two important subtasks of information extraction. The problem of entity identification from biomedical text corpora has been found to be much harder than the identification of entities in areas such as economics or news wire service. NER is the first step towards relation extraction. RE, as a second step, deals with the problem of finding associations and roles between entities within a text phrase. Only little work for biomedical relation extraction exists so far and almost all of the methods focus on co-occurrence approaches or try to learn rules to extract relations. This thesis, in contrast, focuses on methods from the machine learning approach.

We tackle the named entity recognition problem by using a single probabilistic model called Conditional Random Fields (CRFs), recently introduced by Lafferty et al. [2001]. In particular, we did not want to use any application specific pre- or post-processing and therefore wanted to focus on the suitability for performance relevant applications. We conduct our experiments on recognizing several biomedical entities and also made an analysis for the CRF approach on a non biomedical domain (newswire domain) to test the generalizability of the approach. It is concluded that Conditional Random Fields are a suitable probabilistic model for named entity recognition for various domains and different languages without using any post-processing. Furthermore, the experiments have all been conducted with the same type of features.

We chose protein-protein interaction extraction as use case for the relation extraction problem. This use case is very interesting, since the number of protein interactions reported in literature is substantial and growing rapidly. Additionally, the study of protein-protein interactions is a significant technique for studying protein functions, one major challenge in molecular biology nowadays. We consider the problem of relation extraction from a supervised point of view and approach the problem with the help of Support Vector Machines (SVMs), introduced by Vapnik [1998]. We use a rich set of features from text and combine them with external knowledge features (ontology features) available for the protein entities. In addition, we use different views of describing a relation, an idea recently motivated by Xu et al. [2006]. Again, we can conclude that supervised methods are a suitable choice for the RE task. The use of ontology features can improve recall, though the use of entity features could not help to boost performance significantly in these experiments.

Most of the methods and demos developed in the thesis are developed in the object-oriented programming language Java, which may be used for extended analyses in this area.

Zusammenfassung

Diese Arbeit befasst sich mit Named Entity Recognition (NER) und Relation Extraction (RE) aus biomedizinischen Texten. Die zwei eben genannten Problemstellungen sind zwei wichtige Teilgebiete der Informationsextraktion (engl. *Information Extraction*). Das automatische Erkennen von biomedizinischen Entitäten aus Texten hat sich als weit schwieriger erwiesen als das Erkennen von Entitäten, die typischerweise in der Wirtschaft oder in Zeitungen zu finden sind. NER ist dabei sogleich der erste Schritt zur Lösung des RE Problems. Nachdem die Entitäten erfolgreich aus einem bestimmten Teil eines Textes identifiziert wurden, befasst sich RE mit dem Auffinden von Verbindungen zwischen den Entitäten und deren Rolle, die sie untereinander spielen. Bisher gibt es nur wenige Arbeiten, die sich mit dem Auffinden von Relationen zwischen biomedizinischen Entitäten aus Text befasst haben. Der Großteil der Arbeiten beschäftigt sich allerdings nur mit Ansätzen, die entweder die Häufigkeit des gemeinsamen Auftretens zweier Entitäten untersuchen oder aber Regeln für das Extrahieren von Relationen definieren. Diese Arbeit dagegen konzentriert sich ausschließlich auf maschinelle Lernmethoden.

Wir benutzen Conditional Random Fields (CRFs), die erst kürzlich von Lafferty et al. [2001] eingeführt wurden, um Entitäten automatisch zu identifizieren. Wir verzichten auf domain-spezifische Vor- oder Nachbereitung und legen dabei besonderen Wert auf die Performance unserer Lösung. Wir versuchen mehrere biomedizinische Entitäten zu identifizieren. Zusätzlich führen wir Experimente mit CRFs in einer nicht biologischen Domäne (Zeitungsdomäne) durch, um auf die Anwendbarkeit von CRFs in anderen Domänen schließen zu können. Wir schlussfolgern, dass CRFs geeignete probabilistische Modelle sind, um Entitäten in unterschiedlichen Domänen und Sprachen zu erkennen. Darüber hinaus ändern sich die Merkmale über die diversen Domänen und Sprachen hinweg nicht.

Die Identifizierung von Protein-Protein Interaktionen wird als Anwendungsfall für die Extraktion von Relationen gewählt. Dieser spezielle Anwendungsfall ist von besonderem Interesse, da die Anzahl der Proteininteraktionen, die in der medizinischen Literatur veröffentlicht werden, beträchtlich ist und schnell zunimmt. Zusätzlich ist das Wissen über bekannte Proteininteraktionen ein Schlüssel zur Erkenntnis über die genaue Funktion der Proteine, eine der heutigen Herausforderungen der Molekularbiologie. Wir betrachten das RE Problem aus einer überwachten Perspektive und wenden Support Vector Machines an, die von Vapnik [1998] eingeführt wurden. Dabei benutzen wir sehr heterogene Merkmale. Einerseits werden diese Merkmale aus Text extrahiert und andererseits machen wir uns bereits bekanntes Wissen (u. a. Ontologien) zunutze, um die Proteinentitäten zu charakterisieren. Wir adaptieren eine Idee von Xu et al. [2006] und beschreiben Relationen aus verschiedenen Perspektiven. Wir folgern erneut, dass überwachte Methoden geeignet sind, Relationen aus Text zu extrahieren. Ontologiemerkmale scheinen dabei hilfreich zu sein, wobei der Gebrauch von Entitätsmerkmalen die Performance nicht signifikant verbessern kann.

Die meisten Methoden und Demonstrationen wurden in der Programmiersprache Java geschrieben und sind für weitere Analysen auf diesem Themengebiet geeignet.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis focuses on the problem of information extraction (IE) from text in the bioinformatics area. Information extraction deals with the search for entities, relationships among them, or other fact-statements within an article or a certain text phrase. IE can be seen as a subfield of text mining, an area which has gained a lot of patience not only in the biomedical area in the last decade. The importance and the background of information extraction is highlighted in section 1.1. Common concepts in this area are also described in this section. Benefits and possible applications to IE are presented in section 1.2. In particular, we focus on two current problems of IE, namely named entity recognition (NER) and relation extraction (RE).

NER seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations and locations. Named entities are phrases or proper nouns and vary through different domains. NER deals with identifying and classifying these phrases to the correct category of a possible set of entities. The problem of entity identification from biomedical text corpora has been found to be much harder then the identification of entities in areas such as economics or news wire service.

RE deals with the problem of finding associations and roles between entities within a text phrase. Relation extraction usually consists of a two step approach. First, the entities are identified with help of NER and second the relationships between them are extracted. This thesis is thereby solely orientated on methods from the machine learning approach.

Machine learning is a broad subfield of artificial intelligence (AI), thus machine learning is concerned with the development of algorithms and techniques that allow computers to 'learn'. Here, we focus on supervised learning. The objective of supervised learning is to learn a function $f : \mathbb{R}^m \rightarrow C$ that is able to assign to a given (unseen) $\vec{x} \in \mathbb{R}^m$ a class label $c_i \in C$. The vector $\vec{x} \in \mathbb{R}^m$ is the feature representation of the instance. Each feature represents one property of the instance. The model or classifier is learned on a set of training instances $(\vec{x_j}, c_i), j \in 1, ..., m, i \in 1, ..., n$ where the diverse c_i are already known from a trusted source (e. g. from a human annotator). The task of learning such a function is referred to classification, if c_i is restricted to a small number of categorical values (classes). Otherwise, the values to be predicted are of quantitative nature and the task is regression. If $n = 2$ the classification is called binary, the most common classification task.

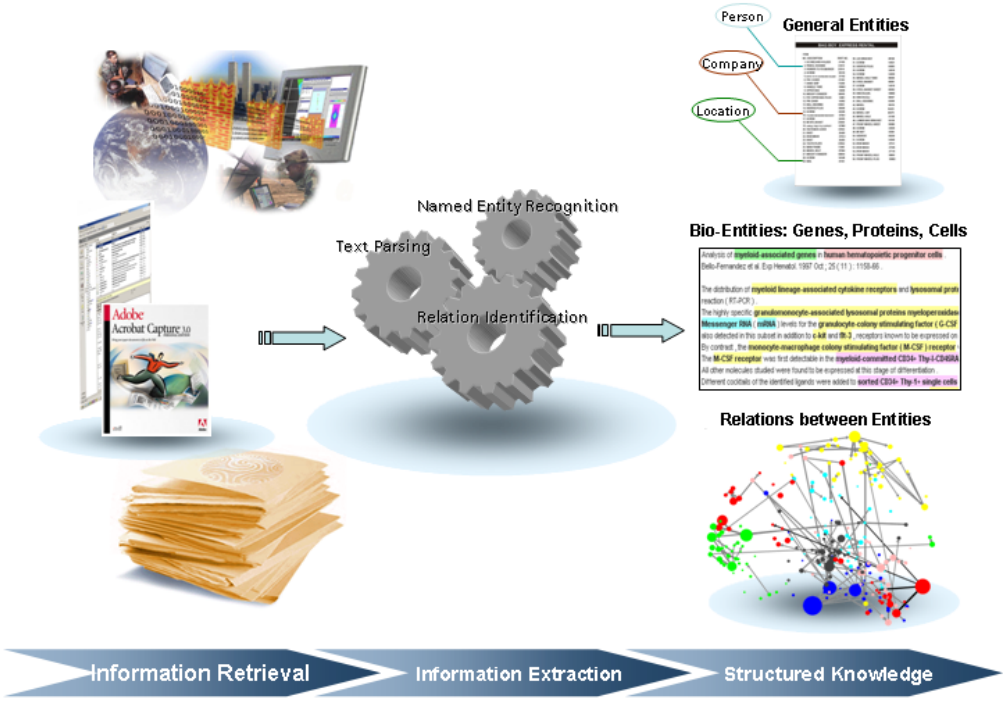

Figure 1.1: Workflow of a text analysis pipeline

If $n > 2$ we are talking about multilabel classification.

We tackle named entity recognition by using a single probabilistic model called Conditional Random Fields (CRFs), recently introduced by Lafferty et al. [2001] (see chapter 4.1). For comparison, a review of the most commonly used methods is presented in Chapter 2. For more details about the experimental settings and results see section 5.1.

We chose protein-protein interaction extraction as use case for the RE problem. We tackle the RE problem with the help of Support Vector Machines (SVMs), invented by Vapnik [1998]. Section 4.2 introduces this machine learning technique. A review of the commonly used methods and the motivation for choosing this use case is presented in chapter 3. The results of our approach are presented in section 5.2. The end of the thesis includes a discussion and outlines future work on this topic (chapter 6).

1.1 Background and Motivation

The last decade has been undergone an unprecedented increase of biomedical data and published literature discussing it. Progress in computational and biological methods have dramatically changed the scale of biomedical research. Large-scale experimental methods constantly produce large quantities of data and even complete genomes can be sequenced within months. This immense volume of data presents a major data analysis challenge.

The explosion of information in biomedical literature and particularly in genetics has highlighted the urgent need for automated text information extraction methods. The published literature can be used as a rich resource of knowledge, since a lot of important information is available only in free text and is not stored in databases yet. The ultimate goal is to exploit this knowledge source efficiently and transfer the incredible bulk of unstructured data (semi-)automatically into a structured form. Therefore retrieving information from free text has become an important subfield of bioinformatics.

Retrieving information from free text is a very vast notion, thus we want to specify and characterize the most important steps of a text analysis pipeline. *Text mining* is a variation of data mining, nevertheless the two concepts share the same objective: to gather and discover yet unknown knowledge. The difference between regular data mining and text mining is that in text mining the patterns are extracted from natural language text rather than from structured databases of facts. This already stresses one major challenge of text mining: While databases are designed for programs to access and analyze the data automatically, text is written for people to read. Sometimes one wants to gather knowledge only on a very specific subset of a large collection of documents a use case where the so-called *Information Retrieval* (IR) fits in.

Information retrieval from text aims at finding the relevant subset of documents out of a large database of documents given a specific information need, usually expressed by a user query. Indexing a document collection is a necessary preprocessing step in order to handle the information retrieval task accurately and efficiently. Indexing can be defined as the process of determining the terms or words within each individual document that should be used when matching a certain document to a query. Mathematical techniques for determining the usefulness of particular words for indexing documents do exist [Salton, 1989]. However, a number of factors make indexing for the biomedical domain more complicated. E. g. , many entities of interest in the biomedical domain, like genes or proteins, have synonym names and no distinctive identifier. Therefore, a disambiguation of these terms for indexing is not a trivial task. Additionally, a large fraction of interesting concepts in this domain have a word length of two or more, making traditional indexing difficult, since this approach assumes that the unit of interest is a single word. (see Cohen and Hunter [2004] for more details on problems with indexing in the biomedical domain). *Document clustering* and *Text categorization* are tasks often addressed by information retrieval systems. Document clustering deals with grouping the documents into a set of clusters according to a some predefined similarity metric, while text categorization deals with the labeling of documents from a predefined set of category tags. The next logical step in a text analysis pipeline, after having performed the IR step, is *information extraction* (IE).

Information extraction deals with the identification of relevant phrases or statements within an article or within a certain text passage. Therefore an IE systems mostly looks for entities and relationships among them (for more details, see section 1.1.2). Dependent of the application one possible last scenario could be the visualization of relationships among the entities across or within a set of documents. Filling templates for a database could be a possible other scenario, e. g. . Figure 1.1 shows a possible workflow for a text analysis pipeline. Note that dependent on the kind of application different steps and end points

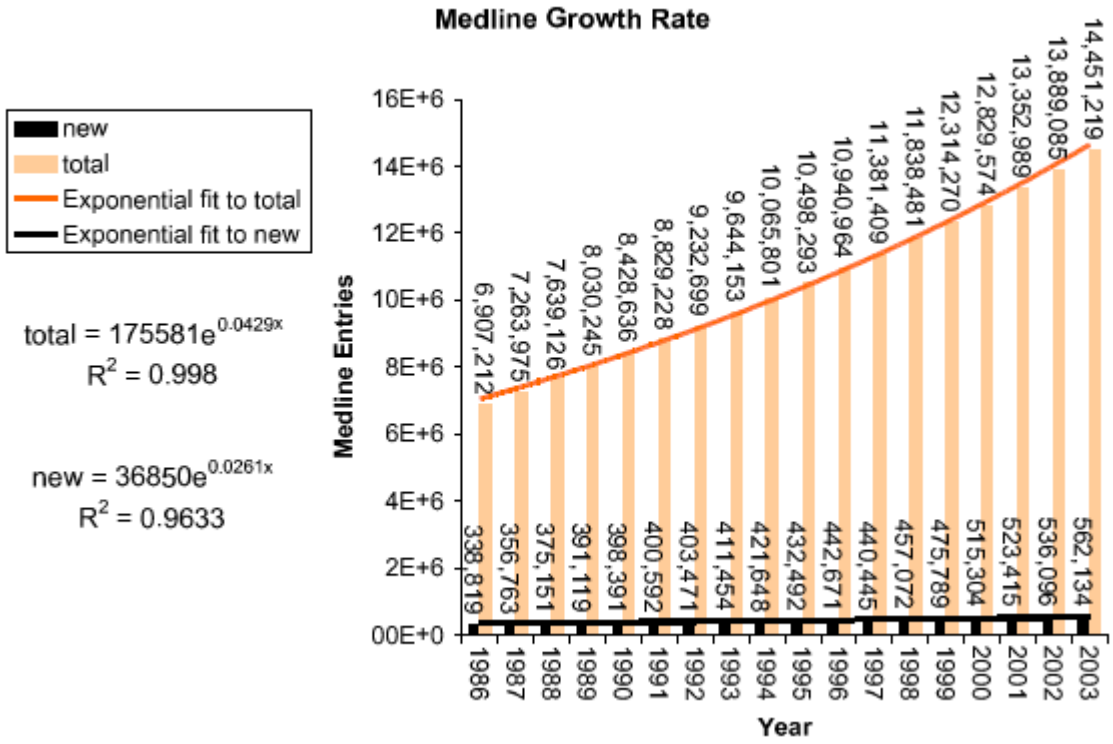

Figure 1.2: Growth of the medline database from 1986 to 2003 from Cohen and Hunter [2004]

of an application can differ. Information extraction often makes use of techniques from Natural Language Processing (NLP). NLP can be defined as the processing of natural language by computers (see the next section 1.1.1 for more details).

1.1.1 Architecture of Information Extraction Systems

Typically, IE systems are comprised of several subunits, where each subunit performs a special task with respect to the extraction of entities and their relationships. IE systems mostly have three to four subunits. Some units, like tokenization, are essential for further extraction steps. However, some units are, dependent on the application, not necessary (e. g. morphological analysis).

- **Tokenization:** The first component is the so-called tokenization, the partition of the document into the basic text blocks. Basic text blocks can be words, sentences or paragraphs, depending on the application. However, the most common form of tokenization for most text mining systems is the splitting of text into words and sentences. Note that there are different tokenization schemes. Different tokenization schemes define tokens in different ways. One scheme could treat 'MMP-related' as

one token while another scheme treats this string as three tokens, namely 'MMP', '-' and 'related'.

- **Morphological Analysis:** After having partitioned the text into the building blocks, a morphological analysis can be performed. The morphological analysis assigns to each word its part-of-speech (POS) tags. POS tags are a set of word-categories, which reflects the role of the word in a sentence. Most systems use 7 different word categories like *Article, Noun, Verb, Adjective, Preposition, Number and Proper Noun*. POS systems are either rule-based or probabilistic. State of the art POS tagger achieve an performance about 94%- 96% accuracy in the general language domain (e. g. newswire domain). Due to the special style of writing in biomedical research publications, performance of general POS taggers will significantly decrease when analyzing biomedical documents. Therefore, a POS tagger has to be trained extra for this domain. See Cohen and Hunter [2004] for a more detailed overview of Part-of-Speech tagging.

- **Syntactic Analysis:** This third subunit builds the connection between the different parts of each sentence and is done either by full parsing or shallow parsing. This kind of parsing is called syntactic parsing. The input of syntactic parsing is a sequence of tokens, sometimes the POS tags are additionally assigned to the tokens. The parser then builds a syntactic parse tree, where the leafs of the tree corresponds to the tokens of the input and the internal nodes represent syntactic structures like *Noun, Verb, Phrase, Noun Phrase* etc. Unfortunately, this process is very time consuming and in general not practical for large text corpora. An alternative to full parsing is shallow parsing. In this process, so-called phrases are assigned to the tokens, therefore this process does not perform a deep syntactical analysis. The phrases are usually: *Noun Phrase, Verb Phrase, Prepositional Phrase, Conjunction Phrase* etc. but the exact phrase set depends on the application. There are some existing experiments, which conclude that only shallow parsing could improve their system performance and therefore suggest to skip the deep parsing procedure [Zhou et al., 2005, Bunescu and Mooney, 2005].

- **Domain Analysis:** This last step combines all the steps mentioned before and extracts the most important information, namely the entities and their relationships. The domain is of course dependent of the kind of application. See figure 1.3 for a possible use case scenario.

Note that anything which goes beyond tokenization is considered as deeper syntactic knowledge in this thesis. Throughout the thesis we don't make use of any sophisticated NLP knowledge.

1.1.2 Information Extraction for Bioinformatics

Scientists in biology or related fields have to cope with an increasing body of literature, which grows too rapidly to be reviewed by researchers alone. Additionally, a lot of im-

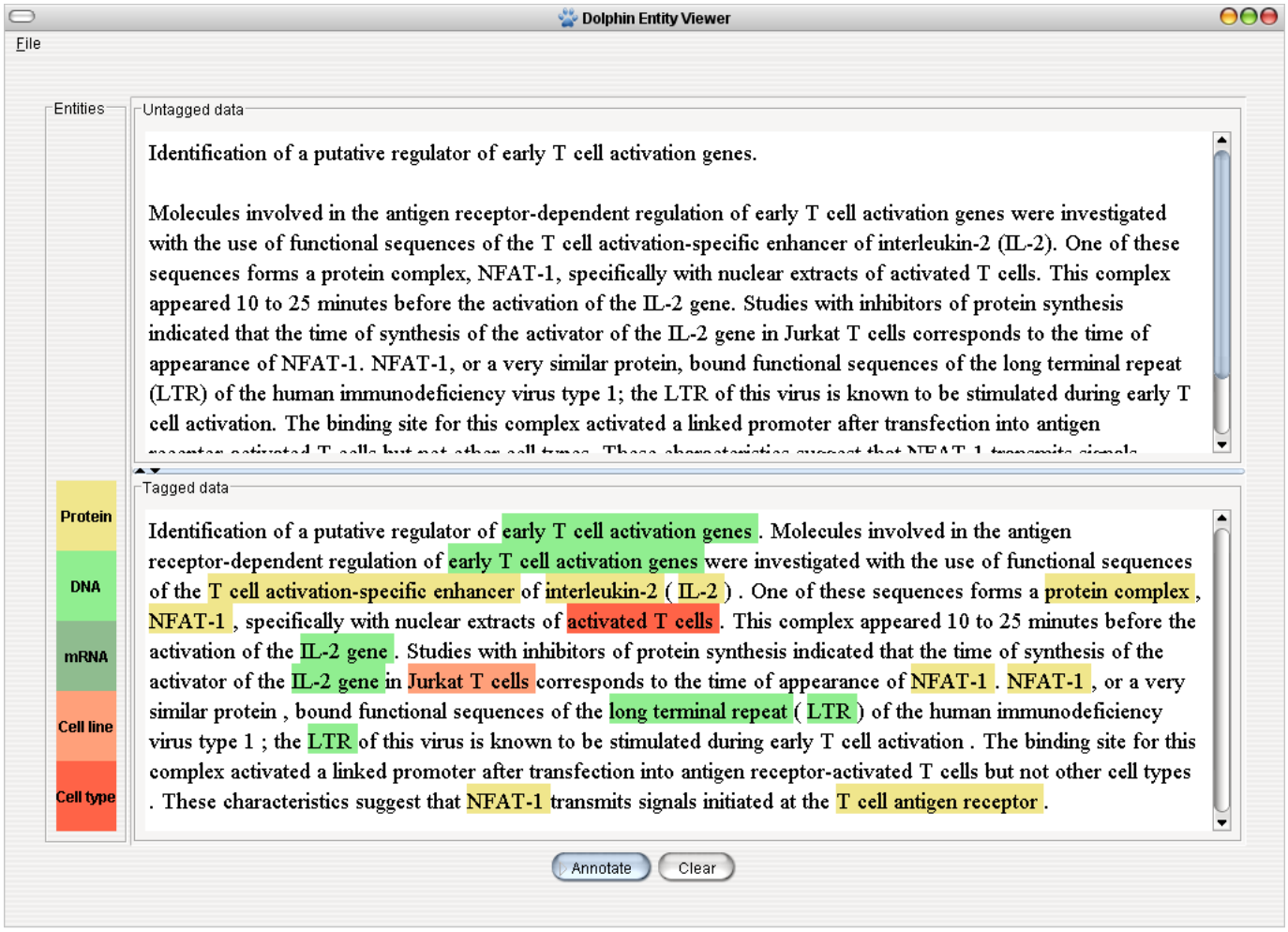

Figure 1.3: Demonstration of our developed NER system. This model was trained to distinguish between five different biomedical entities (see Chapter 5.1 for more information about the model and different experimental settings).

portant information is going to be published outside of the traditional biology subfields (e. g. bioinformatics). Figure 1.2 gives an impression about the increase of publications, researches have to struggle with. Nevertheless, the need for a systematic survey of genes, proteins, diseases, drugs and other chemical compounds still exists and therefore the biomedical community becomes more and more aware, that there is an urgent need for automated exploitation of the literature. Since the early 1990s a number of conferences have picked up topics dealing with text mining or other subfields for mining the biomedical literature. The Pacific Symposium on Biocomputing (BSP), the Critical Assessment of Information Extraction systems in Biology (BioCreAtIvE) and Intelligent Systems for Molecular Biology (ISMB) are conferences publishing papers on this topic.

There are a number of existing information extraction systems for bioinformatics and the level of knowledge to be extracted by the different applications is quite diverse. The objective of information extraction systems for bioinformatics is typically centered around finding information for genes or proteins and their relationships between. Craven and Kumlien [1999] and Ray and Craven [2001] designed systems to identify protein sub-cellular localizations and gene-disorder associations. The systems distinguish between sentences which do include some important facts and sentences which are free from relevant information. Therefore they just classify the sentences according to their information content, rather then extracting facts from the sentences. Rindflesch et al. [2000] designed a system to extract information about drugs and genes within a certain cell relevant to cancer from the biomedical literature. This systems tries to extract facts about the relationships for the just mentioned entities. While these papers try either to distinguish fact-bearing from not relevant information or try to extract some facts, some existing work also tries to extract real protein networks from literature and compares these networks with existing ones from databases [Ramani et al., 2005]. The resulting network from Ramani et al. [2005] extracted from literature was combined with existing networks and resulted in a network with 7748 proteins and 31609 interactions. The interesting use case of extracting protein-protein interactions from text will be reviewed more in detail in Chapter 3.

1.1.3 Evaluation Methods for Information Retrieval and Information Extractions Systems

It is essential to measure the potential merit of a text analysis tool by comparing it to other techniques with respect to some gold-standard. Therefore, we need an annotated, tagged text corpus that represents the gold standard. In order to measure the potential merit in an objective manner, human experts annotate the text corpus and the trade-off of the annotators constitutes the standard and the so-called inter-annotator agreement. Thus, the inter-annotator agreement describes an upper bound for the best possible performance of the text analysis system. In order to be able to measure how well our text analysis tool performs with respect to the gold standard, we need to define a distance measure. An usual way to address the performance measurement in information retrieval and information extraction systems is in terms of *recall* and *precision*. Let us denote **N** as the number

of items, that a text analysis system has to label as 'correct' or 'false' according to some criterion. Items can be either documents, sentences or terms and the criterion could be a membership in a certain document class, a user query or an entity class. Items labeled correctly as positive represent the number of *true positives* (TP), while the number of items labeled correctly as negative are named *true negatives* (TN). Items which have been wrongly assigned to the 'correct' class are called *false positives* (FP). Items where the systems failed to classify them as 'correct' are denoted as *false negatives* (FN). Altogether, these four item sets sum up to **N**. *Precision* (P) is now defined as the fraction of TPs with respect to all items that are labeled as 'correct':

$$P = \frac{TP}{TP + FP}. \tag{1.1}$$

Recall (R) is the fraction of true positives with respect to all items which should have been labeled as 'correct':

$$R = \frac{TP}{TP + FN}. \tag{1.2}$$

A text analysis tool should have high-performance in terms of precision and in terms of recall . A measurement, which combines the two terms is the so-called F-measure. The traditional F-measure or balanced F-score is the harmonic mean of precision and recall:

$$F_1 = \frac{2PR}{P + R}. \tag{1.3}$$

This evaluation method is the common standard way of measuring performance for both, the named entity recognition problem and the relation extraction problem.

1.2 Goals of Entity - and Relation Mining

As already mentioned in previous sections, due to the vast amount of information stored in free text, the need for automated processing of literature is apparent. In this section, we want to outline possible applications and benefits, resulting from information extraction. We focus on high-level applications rather than listing all the range of applications for different domains and different scenarios. Some domain-independent goals of mining entities and relations are:

- **Automatic creation and curation of databases and knowledge bases:** Especially in the biomedical area, curating databases is very time consuming and requires intensive work by human experts. Having solved the named entity and relation extraction problem accurately, one could easily help human curators to fill in database templates.
- **Question Answering:** Information extraction could help to solve the question answering problem. Event based summarization can give answers to question like 'Who did what and when?'. However, this is a very challenging task, since it requires understanding of human language. Thus, this is still a very open research area.

- **Additional knowledge source for IR:** It could be extremely helpful for IR systems to have additional knowledge about entities and relationships stated in certain documents as facts. These facts could act as kind of higher level features and could improve the performance of IR systems significantly.

- **Simplified navigation through text:** Entities can be important pointers to very informative regions of a document. Therefore, simple highlighting of entities or extracting fact statements can already point the reader to the essential phrases of the document.

Of course, the just mentioned applications hold for bioinformatics applications, too. Additionally, two other important items where IE (and thus entity- and also relation extraction) fits into the bioinformatics analysis pipeline have to be mentioned:

- **Support of analysis of high-throughput assays:** Microarray expression profiling produces large amounts of data and it is a challenging task to interpret these data. Karopka et al. [2004] , e. g. , address this problem and try to integrate existing knowledge from literature into the analysis process to create a combined network of genes. Therefore, they use information extraction techniques to extract protein relations from text and analyse the resulting interactions with the interactions referred from the microarray data. Using literature as external knowledge support is one way to fit information extraction into the gene expression pipeline. Another goal of information extraction (but also of information retrieval, of course) is to provide researchers with tools which can navigate them efficiently through literature when analyzing gene expression array results. Entity- and relation extraction can contribute a lot to this task.

- **Construction of interactions and pathways:** A lot of interactions are only reported in literature and not yet stored in databases. Therefore, IE from biomedical literature is a valuable source for constructing pathways and can also uncover very interesting interactions. Ramani et al. [2005] used biomedical literature mining to consolidate the set of known human protein-protein interactions. They first applied machine-learning based named entity recognition and afterwards they applied a co-occurence based approach to extract possible interactions. Additionally, these interactions were filtered for physical interactions with help of a bayesian classifier. In total, they mined 31609 interactions among 7748 human proteins. Since more than 375000 interactions are expected in the complete human gene network [Ramani et al., 2005], the extracted network reveals only about 10% of the complete network. Again, entity- and relation extraction play a crucial role in this task.

Chapter 2

Named Entity Recognition (NER)

This chapter gives an introduction to the problem of recognizing named entities from text. We point to current state of the art techniques for solving this problem and highlight the importance of named entity recognition for biomedical texts. In addition, we describe the problem of NER as task of labeling sequential data to motivate the approach of conditional random fields.

NER is an important subtask of information extraction and once this problem is solved more complex mining tasks can be addressed (e. g. , relationship extraction). The problem was first defined in the general-language domain in the context of the Message Understanding Conferences [Grishman and Sundheim, 1995]. Named entities are phrases or proper nouns and vary through different domains. NER deals with identifying and classifying these phrases to the correct category of a possible set of entities. NER can be divided into two subtasks, term identification and term classification. Identification finds the region and the boundaries of an entity and term classification deals with the assignment of a certain class to an entity. Named entities can be the names of persons, organizations and locations. Example: '***Google Inc.*** CEO ***Eric Schmidt*** joins ***Apple's*** Board of Directors'. This sentence contains three named entities. ***Google Inc.*** and ***Apple*** are instances of the concept company, while ***Eric Schmidt*** is an instance of the concept person. In the context of biomedical text, named entities can be proteins, genes, cell-lines, cell-types, diseases, drugs and many more. Most of the entities can usually be identified and unambiguously be classified by human annotators. However, there are cases, where even human annotators have problems to assign a category clearly to an entity. As already mentioned in Section 1.1.3 the inter-annotator agreement measures to which extent the different human annotators assign the same categories to entities. Note that inter-annotator agreements for NER tasks differ from domain to domain. Inter-annotator agreement for newswire text is estimated to be about 97% [Grishman and Sundheim, 1995], while the agreement for biomedical text is about 87% [Hirschmann]. Thus, the inter-annotator agreement can be seen as an indicator of the difficulty of the NER task.

Biomedical entity recognition, where inter-annotator agreement is below 90%, is obviously not a trivial task. The problem of recognizing the named entities of biomedical entities like genes is challenging for several reasons. When not curating a dictionary ex-

tensively, one quickly becomes aware that simple text matching against a dictionary does not suffice, since first of all there are no existing complete dictionaries for bio-entities. In addition, a word can change its entity depending upon context (e. g. ferritin can be referred as biological substance or as laboratory test). Furthermore many biological entities have several names (e. g. PTEN and MMAC1 refer to the same gene) and biological entities may also have multi-word names (carotid artery), so there is an essential need for sophisticated methods. The identification of boundaries for biomedical entities is also quite difficult, since verbs and adjectives can be part of an entity (e. g. mullerian inhibiting substance). But also for newswire text, this seemingly simple task faces some challenges. Entities may be difficult to find, and once found, difficult to classify. E. g. , person names as well as location names can be part of company names.

Before starting with the recognition and classification of named entities, some general preprocessing has to be performed. Usually, the basic text phrases for recognizing entities are sentences. Therefore, if you want to extract all named entities within an paragraph, abstract or article, you have to do sentence splitting first. After sentences are identified tokenization and if needed further morphological analysis can be started (see Section 1.1.1).

2.1 Related Work

Here, we focus on methods and publications for the biomedical NER domain. NER has attracted many researchers and the approaches existing so far for the biomedical domain fall into the two following categories:

- dictionary-based [Krauthammer et al., 2000]
- rule-based [Fukuda et al., 1998]
- machine learning-based [Zhou et al., 2005]

The methods proposed for biomedical NER vary in their degree of reliance on dictionaries, statistical or rule-based approaches. Sometimes it is quite difficult to restrict a system to one of the above mentioned categories. Most approaches require the use of POS tags. POS information can be used as features in machine learning methods, but also for rule-based systems for error discovery or for conditioning rules. According to Cohen and Hunter [2004] dictionary-based approaches perform quite poor in terms of recall (only 10%-30%), even when allowing some different spellings variants between the reference source and the corpus. An exception with respect to low recall is the dictionary-based system of Krauthammer et al. [2000] which relies on BLAST in order to identify protein names.

Rule-based approaches typically define some patterns and some logic to match biomedical entities. In addition, rules are defined for extending names to the right and/or left. Patterns are usually implemented as regular expressions or as combinations of these. '[\W+\d+]' would be an expression to recognize the string representation of a protein like 'MMP12', but would not recognize the representation of 'MMP-12' or 'MMP 12'. One

drawback of rule-based approaches immediately becomes apparent: Since there is no standard biomedical term naming convention for most of the biomedical entities (chemical entities with the IUPAC convention represent one exception), the rule building process is quite difficult as the number of possible alternative spellings increases. Additionally, rule-based approaches are very domain specific and existing rules for one named entity class usually perform quite poor to new classes of named entities. Therefore, setting up rules for several named entities is very time consuming. Fukuda et al. [1998] has set up such a rule-based system named PROPER (**PRO**tein **P**roper-noun phrase **E**xtraction **R**ules). However, the results they present hold only for very specific protein domains and the test set is quite small (only 80 abstracts).

A variety of machine learning algorithms have been tried to entity identification. There are mainly two categories of classifiers, classifier-based and markov model based classifiers. Classifier-based models include SVMs, naive Bayes and decision trees. Markov model based models include hidden markov models (HMMs), maximum entropy markov models (MEMMs) and more recently Conditional Random Fields (CRFs). Markov model based systems are a suitable choice for solving sequence tagging problems such as speech recognition or POS tagging. See Section 4.1 for more information about markov model based systems. The development of the GENIA 3.0 corpus [Kim et al., 2003] has overcome the problem of lack of training data for machine learning based algorithms. GENIA 3.0 contains about 2000 abstracts with diverse biomedical entities and has become a kind of standard evaluation corpus. Many other corpora for evaluation of biomedical NER are derived from the GENIA corpus, such as the BioNLP/NLPBA (see Section 5.1.1) or the BioCreAtIvE corpus(see Section 5.1.2 for further information).

Post-processing has also gained a lot of attention for biomedical named entity recognition. Typical work include Lin et al. [2004] and [Zhou et al., 2005]. Lin et al. [2004] report a 20% performance increase due to their post-processing. They follow a kind of rule-based boundary extension strategy and combine this with a dictionary for reclassification of entities. Zhou et al. [2005] use abbreviation resolution to post-process their initial results. Due to their post-processing strategy they rank first in the BioNLP/NLPBA evaluation. See Section 5.1.1 for a short description of their system.

2.2 NER as Segmenting and Labeling Task of Sequential Data

The identification of entities from text can be treated as a tagging task. Here we regard each word in a sentence as a token. Each token $\mathbf{x_i}$ is associated with a tag or label $\mathbf{y_i}$ which indicates the type of the entity $\mathbf{x_i}$ (e. g. *Google* would be an example of the entity *company*). Additionally $\mathbf{y_i}$ also indicates whether we are outside (**O**) of an entity (a token does not belong to an entity) , at the beginning of an entity (**B-t**) or inside a certain type of entity (**I-t**), where **t** is the type of entity. Therefore a sentence can be divided into its corresponding tokens $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$ and its corresponding labels $\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}$. Two random variables **X**

and $\mathbf{Y}$ are used to denote any input token sequences and its associated label sequences. All components $\mathbf{y_i}$ of $\mathbf{Y}$ range over a finite label alphabet γ. The task is now to assign to a given token sequence $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}$ its correct label sequence $\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}$. Consider the following token sequence: 'IL-2 gene expression and NF-kappa B activation through CD28 requires reactive oxygen production by 5-lipoxygenase . '. The corresponding label sequence would be: 'B-DNA, I-DNA, O, O, B-protein, I-protein, O, O, B-protein, O, O, O, O, O, B-protein, O'. Note that a token cannot be split between a marked part and an unmarked part. For example, if 'NF-kappa' is a token and one wants to mark 'NF' as part of an entity, then one also need to mark '-kappa' as part of the entity. For training such labeled token sequences are given, for testing the systems takes as input the tokenized unannotated sentences. We use this perspective of NER to solve the problem with the help of Conditional random fields (see Section 4.1).

Chapter 3

Relation Extraction (RE)

This chapter introduces the task of extracting relations from text. We summarize related work for extracting protein-protein interactions and highlight this use case.

Reliably extracting relations between entities in natural-language documents is a very difficult, still unsolved problem and most of research on text information extraction has focused on tagging named entities. Nevertheless, the next logical step for IE is to begin to develop methods for extracting meaningful relations involving named entities. As already mentioned earlier, such relations would be extremely useful in applications like question answering, automatic database generation and intelligent document searching and indexing. The Automatic Content Extraction (ACE) program is the only conference which addresses the relation extraction problem and provides an annotated benchmark set for relations holding between entities like, e. g. , persons, organizations and locations. ACE provides five types of relations, which can be divided in several subtypes. Example: '*TWA* pilot *Steve Snyder*.' This sentence has two entities, namely, TWA and Steve Snyder and between these two entities a *member* (subtype) relation and automatically a *role* (since the role relation is a higher-order relation of the member relation) relation holds. So the task in ACE is not only to predict whether there is a relation between two entities or not, you also have to assign a category to the relation. Therefore relation extraction can be divided into two subtasks. The first one is the mere detection of relations and the second one also includes the characterization of the relation. At ACE this whole process is referred to as Relation Detection and Characterization (RDC). While Zhou et al. [2005] report quite satisfying results for relation detection (74.7% F-measure), the results for RDC are significantly lower (68.0% F-measure).

In biomedical relation extraction most of the existing work focuses on relation detection. Only very little work has been done on the RDC task. Only Rosario and Hearst [2004] have presented some work on this topic. They use generative graphical models and one discriminative model to extract seven different relations that hold between diseases and treatments. In addition, they conduct experiments on extraction of several kinds of protein-protein relations [Rosario and Hearst, 2005]. The remaining existing work focuses on relation detection. Hereby, the different kind of entities which are involved vary.
Craven and Kumlien [1999] focus on detecting associations between proteins and sub-

cellular locations and Rindflesch et al. [2000] extracts relations holding between genes, drugs and cell-lines related to cancer. However, most of existing work focuses on extracting relations between proteins. The study of protein interactions has been vital to the understanding of how proteins function within the cell and extensive protein interactions maps do already exist for yeast, worm and fly but not for humans. Ramani et al. [2005] compared existing interaction databases (BIND, DIP and HPRD) and discovered that these existing data sets are quite disjoint. They conclude that the databases are biased for certain classes of protein interactions and in addition, the number of interactions in Medline is quite large. Thus, a lot of interactions are only reported in free text and retrieving these protein interactions manually is not tractable, due to the large number of articles.

3.1 Related Work

In this section we outline existing work on protein-protein interaction extraction from text. Current research on this task can be classified into the following categories:

- rule-based [Ono et al., 2001]
- co-occurence based [Ramani et al., 2005]
- kernel based [Bunescu and Mooney, 2005]
- other machine-learning based [Rosario and Hearst, 2005]

The major problem of protein interaction extraction is, that no common standard benchmark set does exist (in contrast to NER where a number of benchmark sets do exist). Therefore, it is very difficult to compare performance of different systems. Most of the existing systems suffer from low recall.

Ono et al. [2001] start with a given list of interaction verbs (e. g. interact or bind). With these trigger words and some additional information, they construct hand-built specialized templates. They extract the pairs of proteins which occur with these trigger words in a text phrase. Thus, they can only extract certain types of interactions and they have problems with complicated sentences, where a lot of proteins are stated. In general, a rule can be seen as a sparse subsequence of words or POS tags anchored on two protein-name tokens. [Bunescu and Mooney, 2005] want to overcome the limitation of rule-based systems that are able to extract only a small subset of all possible subsequences of words. Therefore they design a generalized kernel for relations, which relies on patterns for relations. The kernel works with patterns designed for words and word classes (e. g. POS tags). They can show, that this kernel performs better than their implemented rule-based approaches. Ramani et al. [2005] use a two-step strategy to identify interactions. First, they measure co-citation of proteins in Medline and count the number of co-citations of certain proteins and compare them by calculating the probability of this co-citation frequency under a random model. Second, they use a Bayesian filter in order to enrich the extracted pairs for physical interactions. Co-occurence based approaches usually suffer from low precision, since despite

filtering steps, a lot of protein pairs are co-cited very often for many other reasons than physical interactions. As stated before, Rosario and Hearst [2004] first introduced an RDC task in the biomedical domain. Additionally, they report an accuracy of 64% for a 10-multi-way relation classification task on protein-protein interactions when using a neural network [Rosario and Hearst, 2005].

Chapter 4

Methods

4.1 Conditional Random Fields

Conditional Random Fields [Lafferty et al., 2001] are a framework for building probabilistic models for segmenting and labeling sequence data. Conditional Random Fields have outperformed Hidden Markov Models (HMMs) on a number on real-world applications for several reasons which will be presented in the next chapter (see 4.1.1, page 33). Formally a CRF can be defined as an undirected graphical model. In general, graphical models are graphs where the vertices represent random variables and edges represent conditional independence assumptions. For undirected graphical models the absence of an edge between two vertices in a graph $\mathbf{G}$ implicates that the random variables represented by the vertices are conditionally independent. Given two random variables $\mathbf{X}$ and $\mathbf{Y}$ in what follows, is a formal definition of a CRF [Lafferty et al., 2001]:

Definition 1. *Let* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ *be a graph such that* $\mathbf{Y} = (\mathbf{Y_v})_{\mathbf{v} \in \mathbf{V}}$*, so that* $\mathbf{Y}$ *is indexed by the vertices of* $\mathbf{G}$*. Then* $(\mathbf{X}, \mathbf{Y})$ *is a conditional random field in case, when conditioned on* $\mathbf{X}$*, the random variables* $\mathbf{Y_v}$ *obey the Markov property with respect to the graph:* $\mathbf{p}(\mathbf{Y_v}|\mathbf{X}, \mathbf{Y_w}, \mathbf{w} \neq \mathbf{v}) = \mathbf{p}(\mathbf{Y_v}|\mathbf{X}, \mathbf{Y_w}, \mathbf{w} \sim \mathbf{v})$*, where* $\mathbf{w} \sim \mathbf{v}$ *means, that* $\mathbf{w}$ *and* $\mathbf{v}$ *are neighbors in* $\mathbf{G}$*.*

The structure of graph $\mathbf{G}$ represents the conditional independence assumptions made for the label sequences. In theory, the structure of the graph may be arbitrary, but when modeling sequences, the most simple and most common graph is the graph which obeys the first-order Markov property for each random variable $\mathbf{Y_v}$. This means that each label variable $\mathbf{Y_i}$ and $\mathbf{Y_{i+1}}$ are associated in the graph $\mathbf{G}$. Then $\mathbf{Y}$ is said to be a linear-chain CRF. Figure 4.1 shows such a graph. Modeling higher-order Markov models is much more expensive in terms of training time and memory and often requires a higher amount of training data to obtain reasonable results. Remember that the random variables of $\mathbf{G}$ are connected by undirected edges indicating dependencies and let $\mathbf{C}(\mathbf{X}, \mathbf{Y})$ be the set of cliques of $\mathbf{G}$. By the theorem of Hammersley-Clifford [Hammersley and Clifford, 1971] the structure of a CRF can be used to define the joint distribution over elements $\mathbf{Y_v}$ of $\mathbf{Y}$ as

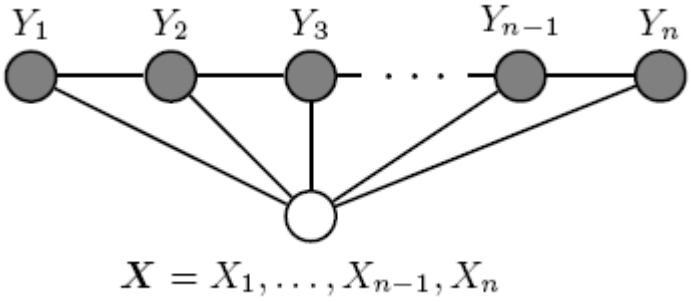

Figure 4.1: Graph G of a linear-chain CRF. G is globally conditioned on the input sequence X. Note: The unshaded variables are not generated by the model. (Figure from Wallach [2004])

a normalized product of strictly positive, real-valued potential functions.

$$P(y|x) = \frac{1}{Z_x} \prod_{c \in C(y,x)} \Phi_c(y_c, x_c), \tag{4.1}$$

where $\Phi_c(y_c, x_c)$ is a potential function and Z_x is a normalization factor. According to Lafferty et al. [2001] each potential function has the form $\Phi_c(y_c, x_c) = e^{\sum_{k=1}^{K} \lambda_k f_k(y_c, x_c)}$, where f is a so-called feature function and λ is a learned weight for each feature function. This form is heavily motivated by the principle of maximum entropy. Maximum entropy is a framework for estimating probability distributions from an incomplete state of knowledge such as training data. The goal of Maximum entropy is to obtain from the empirical distribution and a set of constraints an accurate representation of the process we want to model. According to the principle of maximum entropy, this is the model which is consistent with all the facts and constraints from the training data but otherwise is as uniform as possible [Jaynes] (see section 4.1.2).
Remember our problem description from 2.2 and let now $\mathbf{x} = x_1, x_2, ..., x_N$ be some observed input data sequence (e. g. a sentence, which is a sequence of words in a text document). Let $\mathbf{Y}$ be a set of finite labels of γ and let $\mathbf{y} = y_1, y_2, ..., y_N$ be a sequence of labels. In the remainder of the chapter we will only consider the case of linear-chain CRFs for simplicity reasons. When we are considering the case of linear-chain CRFs the cliques of the graph are restricted to have pairs of states (y_{t-1}, y_t) which are neighbors in the graph. Therefore the conditional probability of a label or state sequence given an input sequence is defined as

$$P(y|x) = \frac{1}{Z_x} \exp\left(\sum_{i=1}^{N} \sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, \mathbf{x}, i) \right), \tag{4.2}$$

where Z_x is the same normalization factor mentioned before, $f_k(y_{i-1}, y_i, \mathbf{x}, i)$ is an arbitrary feature function and λ_k is a learned weight for each feature function and can range between

$-\infty$ to ∞. Each feature function f_i specifies an association between the token that hold at a position and the label for that position. Therefore with each feature function we want to express some characteristic of the empirical distribution of the training data that should also be true for the model distribution. In this work we define only binary features, for instance:

$$f_k(y,x) = \begin{cases} 1 & \text{if WORD= gene} \in x, \text{label}_{-1}(y) = \text{B-Gene}, \text{label}_0(y) = \text{I-Gene}; \\ 0 & \text{otherwise} \end{cases}$$

The corresponding feature weight λ_k specifies whether the association should be favored or disfavored. Higher values of λ make their corresponding label transitions more likely, so the weight λ_k in the example above should be positive, since the word gene will often occur in a gene name. In general, the weight for each feature should be high, if the feature tends to be on for the correct labeling. It should be negative, if the feature tends to be off for the correct labeling and it should be around zero if it is uninformative.
The normalization factor Z_x is the sum over all possible state or label sequences S^N, while N is the length of the input sequence,

$$Z_x = \sum_{s \in S^N} \exp\left(\sum_{i=1}^{N} \sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, \mathbf{x}, i) \right). \tag{4.3}$$

Note that the number of state sequences is exponential in the input sequence length N, therefore calculating the normalization factor for arbitrarily-structured CRFs is intractable and approximation methods such as Gibbs sampling must be used to obtain a reasonable normalization factor. Fortunately, since we are modeling sequences and are dealing the most time with linear-chain CRFs, the so-called partition function (the name for the normalization factor) can be computed efficiently as described in section 4.1.2.

4.1.1 Hidden Markov Models and Related Models vs. Conditional Random Fields

There are several reasons, why we chose CRFs for the NER task instead of HMMs or some other related models like Maximum Entropy Markov Models (MEMMs). First of all, HMMs are a kind of generative model, assigning a joint probability to model observation and label pairs rather than modeling a conditional probability like MEMMs and CRFs. In order to define a joint distribution, generative models must enumerate all possible observation sequences, sth. that for most domains, is intractable unless observation elements are represented as isolated units, independent from the other elements in an observation sequence [Wallach, 2004]. Therefore it is not possible to represent observations in terms of multiple interacting features and long-range dependencies between observation elements.

As can be seen from figure 4.2, HMMs waste effort on modeling the observation sequence when training is performed. MEMMs define the probability of the current label Y_i dependent from the previous label Y_{i-1} and the current input sequence X_i. CRFs are more

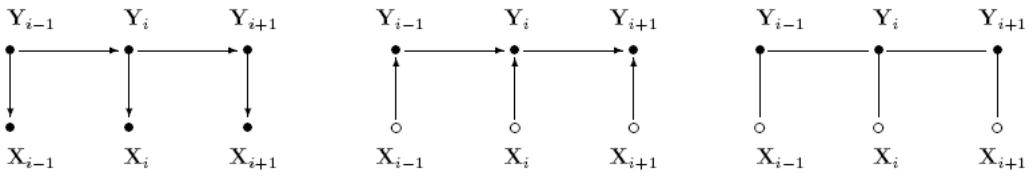

Figure 4.2: Graphical structures of HMMs (left), MEMMs (middle) and CRFs (right) from Lafferty et al. [2001]. Open circles indicate that the variable is not generated by the model.

complex, they are globally conditioned on the input sequence and are dependent on the label sequences which belong to the same clique, in the case of figure 4.2, Y_i is dependent of Y_{i-1} and Y_{i+1}. Additionally CRFs are globally normalized, while MEMMs are locally normalized for every possible state. Another advantage using conditionally models is, when using feature functions, these functions may ask powerfully arbitrary questions about the input sequence, e. g. about previous words, next words and conjunctions of all these.

4.1.2 Parameter Estimation for Conditional Random Fields

We assume that the training data $\mathcal{T}$ is i. i. d. (independently and identically distributed). Thus we define the log-likelihood of the training data to be

$$\mathcal{L}(\mathcal{T}) = \sum_{(y,x)\in\mathcal{T}} \log P(y|x). \tag{4.4}$$

This equation yields some problems. When features rarely occur in the training set and if we assume that these features additionally mostly occur with a special label combination, then this feature will get a very positive weight , which is obviously not desirable. Therefore this feature would contribute a lot to assign a special label. To avoid this overfitting for rarely occurring features, Lafferty et al. [2001] adds a spherical Gaussian prior over features weights and penalizes the likelihood:

$$\mathcal{L}(\mathcal{T}) = \sum_{(y,x)\in\mathcal{T}} \log P(y|x) - \sum_i \frac{\lambda_i^2}{2\sigma^2}. \tag{4.5}$$

Lafferty et al. [2001] shows that the log-likelihood function 4.5 generalizes the well-known case of logistic regression and is concave. Lafferty compute its partial derivatives with respect to the weights

$$\frac{\partial\mathcal{L}(\mathcal{T})}{\partial\lambda_i} = \tilde{E}[f_i] - E[f_i] - \frac{\lambda_i}{\sigma^2}, \tag{4.6}$$

where $\tilde{E}[f_i]$ is the empirical feature count and $E[f_i]$ is the model expectation of feature f_i. The empirical expectations are calculated by simply counting the number of times each feature occurs in the training data. To compute the model expectation is not so

straightforward, the expectation is given by:

$$E[f_i] = \sum_{(y,x)\in\mathcal{T}} \sum_{y'} P(y'|x) \sum_{j=1}^{n} f_i(y_{j-1}, y_j, x_j), \tag{4.7}$$

where $\mathbf{y}'$ stands for all possible labels sequences for the token sequence $\mathbf{x}$. The number of possible tag sequences is exponential on training instance length, therefore computing this sum directly is impractical. However, the Markovian structure of the CRF allows us to use an efficient forward-backward algorithm to compute expectations over label sequences [Lafferty et al., 2001]. For maximizing the log-likelihood function iterative scaling techniques [Darroch and Ratcliff, 1972] and gradient-based techniques [Malouf, 2002] can be used.

4.2 Support Vector Machines

Support Vector Machines (SVMs) are a supervised learning technique first introduced by Vapnik [1998]. Based on the structural risk minimization of statistical learning theory, SVMs seek an optimal separating hyperplane between observations of two classes. In the training phase the model identifies the observations lying closest to the class-separating hyperplane (the support vectors) and assigns weights to these. For the linearly separable case, the support vector algorithm simply looks for the hyperplane with the largest margin (see figure 4.3). Once a support vector machine is trained, we just look on which side of the decision boundary a given test data point x lies and assign the corresponding label. In the case, that the data is not separable (i. e. there exists no optimal hyperplane which can classify all observations correctly), a positive slack variable has to be introduced. This slack variable assigns a kind of extra cost for errors and this cost parameter C is chosen by the user. A smaller C corresponds to assigning a lower penalty on errors. There are classification problems which are non-linearly separable, but there exists a solution for this kind of problem: The so-called kernel trick computes the inner product of two observations in a higher dimensional space. Thus, the hyperplane also lies in this higher dimensional space, permitting non-linear class separation.

Linear, polynomial and gaussian functions are the most popular kernels. Kernel functions can be self-designed, as long as they obey a few mathematical properties (Mercer's condition). Mathematically, as long as the kernel function is symmetric and the kernel matrix formed by the function is positive semi-definite, it forms a valid dot product in an implicit Hilbert space. In this implicit space, a kernel can be broken down into features, although the dimension of the feature space could be infinite. Kernel functions can become so complex that they are able to separate almost any two classes, which will result in a high number of support vector machines. Generalizability of a trained support vector machine correlates with the number of support vectors. A large number of support vectors will decrease generalizability and may result to overfitting of the training observations [Zhang et al., 2006]. SVMs are a popular classifier due to their ability to handle high-dimensional

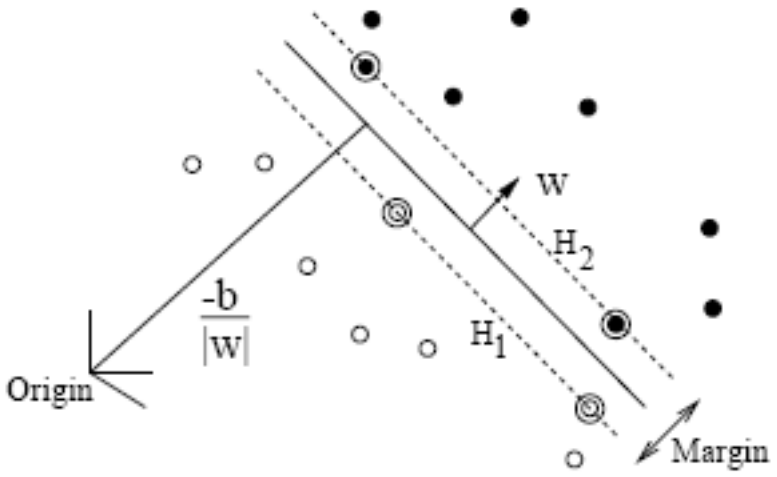

Figure 4.3: Linear separating hyperplane for the separable case from Burges [1998]. Support vectors are circled.

data. Feature weighting is intended to reduce the influence of irrelevant features [Hardin et al., 2004].

Chapter 5

Results

5.1 Named Entity Recognition

In this section, we present the results for named entity recognition. We evaluate the approach of CRFs on four datasets of four different evaluation conferences:

- CoNLL-03 Language-Independent Named Entity Recognition [Erik et al., 2003]
- MUC-6 Named Entity Recognition Task [Grishman and Sundheim, 1995]
- BioNLP/JNLPBA Bio-entity Recognition Task [Kim et al., 2004]
- BioCreAtIvE Task 1a - Gene Entity Recognition Task [Yeh et al., 2005]

In all evaluation conferences the different corpora were given to the participants after some linguistic preprocessing. The level of preprocessing varied through the different evaluations. See the corresponding sections of the conferences for more details on the provided preprocessing level.

The first task covers two languages: English and German. We evaluate our method only on the German dataset. This corpus consists of news stories from the Frankfurter Rundschau from August 1992. The evaluation aims at identifying the entities: persons, locations, organizations and names of miscellaneous entities, that do not belong to the previous three groups.

The second corpus consists of Wall Street Journal texts from the period of January 1993 through June 1994. The original task is to identify persons, locations, organizations, temporal expressions (namely direct mentions of dates and times) and number expressions (namely direct mentions of currency values and percentages).

The third and the fourth dataset are from the biomedical domain and their aim is to identify different biological-motivated entities. At BioNLP/JNLPBA the task is to extract five entities from the given text corpus, namely: Proteins, Dna, Rna, cell line and cell type.

The BioCreAtIvE task on the other hand concentrates on identifying only one entity, namely gene, which in this context means that proteins and genes are merged to this entity. We chose these four datasets for the following reasons:

- The CoNLL-03 and MUC-6 evaluation were chosen to compare the results from the biomedical domain with the results of a newswire domain, where the task of identifying entities seems to be much easier. CoNLL-03 provides a german dataset, so the differences between two languages can be compared. Of course, these differences can not be compared directly, since it is not the same data set, but we would like to encounter some properties of the system when evaluating in different languages.

- The BioNLP/JNLPBA corpus was chosen to simulate a complex entity recognition task, where the different entities are quite similar and difficult to distinguish.

- The BioCreAtIvE corpus is a very new and popular evaluation and can be compared to the CASP (Critical Assessment of Techniques for Protein Structure Prediction) evaluation in the protein structure prediction domain. Besides, with the help of this text corpus we can simulate a much easier task, namely identifying only one bio-entity, namely gene (genes and proteins are merged together in this evaluation).

In section 5.1.1 we introduce the setting of our CRF model and describe the various features we used in detail. Since the model setting and features do not change significantly over the different evaluations we report only the differences of the setting at the other evaluations.

5.1.1 Bio-Entity Recognition Task at BioNLP/JNLPBA

The JNLPBA shared task is thought as an open challenge task and therefore the participants were allowed to use whatever methodology and knowledge sources they liked. The objective of this task is to tag five different biomedical entities.
The training data came from the GENIA version 3.02 corpus [Kim et al., 2003]. The GENIA corpus results from a MEDLINE search using the MeSH (Medical Subject Headings) terms 'human','blood cells' and 'transcription factors'. 2000 (20546 sentences) abstracts were selected from this search and hand annotated with 36 terminal classes. For the shared task only five classes out of the 36 classes were chosen. The test set contains 404 (4260 sentences) abstracts and was obtained from a newly annotated collection of MEDLINE abstracts from the GENIA project. Half of the abstracts of the test data are from the same domain as from the training data, while the other half of the abstracts were extracted from the super-domain of 'blood cells' and 'transcription factors'. The hope behind this choice was to test the generalizability of the methods used. Table 5.1 shows general statistics for the training and test data and figure 5.1 shows the distribution over the word length of the different entities for the test data.
The following rules apply for the scoring. Consider the part of the sentence:' ... a protein related to **SNF1 protein kinase**'.
The bold tokens indicate the markable entities. In order to get a true positive (TP) the system needs to recognize 'SNF1 protein kinase'. If the system marks e. g. 'to SNF1 protein kinase', the system would get a false negative (FN) (for not matching the answer) and

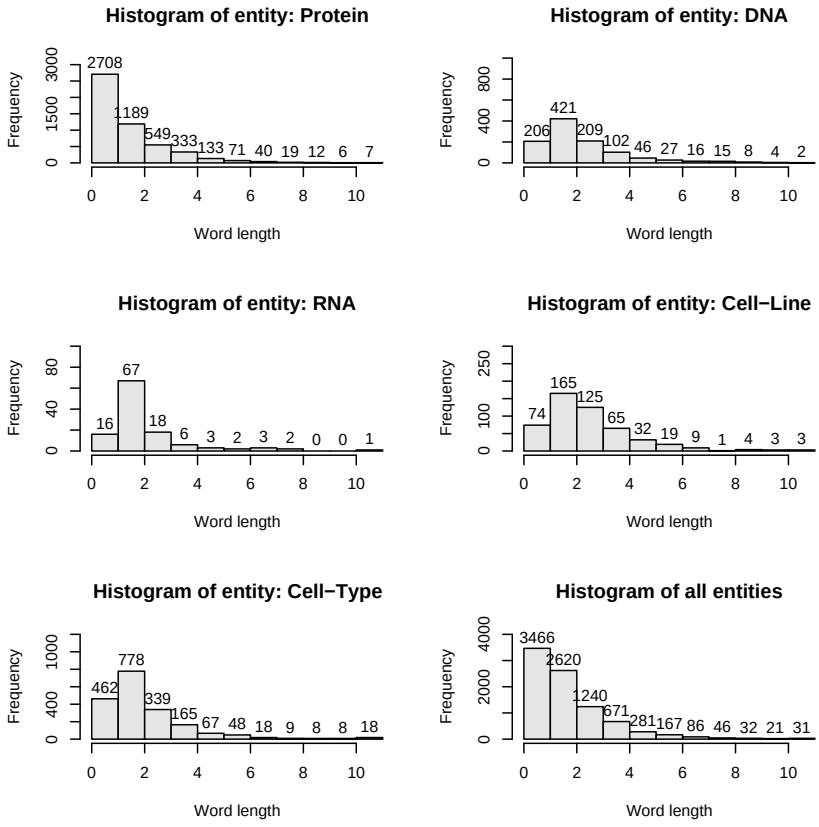

Figure 5.1: Histogram over the word lengths for the different entities of the BioNLP/JNLPBA task. Word length ranges from one to ten, the last bar indicates the words longer than ten.

a false positive (FP)(the returned item does not match the answer). In the following, a description of the features of our CRF approach is given.

Data Set	Protein	Dna	Rna	cell type	cell line	ALL
Training Set	30269	9533	951	6718	3830	51301
Test Set	5067	1056	118	1921	500	8662

Table 5.1: Number of different entities for the training and test set at the BioNLP/JNLPBA task

System description: As already described in section 4.1, the features express some characteristic of the empirical distribution of the training data that should also be true for the model distribution. Therefore, the performance of feature-based models like CRFs strongly depend on the choice of features. This paragraph describes the features we used for our CRF approach. Beside the number of features listed below, we use also the words themselves as features, since they can hold important information of the empirical distribution. We used linear-chain CRFs and second-order CRFs. Figure 5.5 shows the performance of the particular CRFs in respect to the size of training data.

Orthographic Features: Biological entities often consist of capitalized letters, include some numbers or are composed of combinations of both. In addition, genes or protein names have often some greek letters in their name, like NFKP-beta . Therefore it is straightforward to use regular expressions (see chapter 2), which have been shown to be useful in gene mention finding [Mcdonald and Pereira, 2005]. Table 5.2 shows the regular expressions used in the approach of McDonald, which we adopted and slightly modified for our approach. We included some additional regular expressions like expressions which match if a greek letter occurs in a token.

Word Shape Features: Some words belonging to the same class of entities will have the same word shape, e. g. IL-10 and IL-12. Therefore we normalize these words with a very simple approach: Capitalized characters are converted to X, numbers are replaced by 0 , non-capitalized letters are converted to x and non-English characters are converted to _. Thus, normalizing the words IL-10 and IL-12 will result both in XX_00. After applying this normalization, words with similar word shape will be more likely grouped into the same entity class.

N-Gram Features: We also used character n-gram features for $2 \leq n \leq 4$. These features help to recognize informative substrings like 'ase' or 'homeo', especially for words not seen in training. It can be very informative for a token to have a substring like 'ase' or 'prot'. Consider the word 'metallomatrixproteinase', for instance. Additionally, we included prefix and suffix features which can be more informative than the general n-gram

Orthographic Feature	Regular Expression
Init Caps	[A-Z].*
Init Caps Alpha	[A-Z][a-z]*
All Caps	[A-Z]+
Caps Mix	[A-Za-z]+
Has Digit	.*[0-9].*
Single Digit	[0-9]
Double Digit	[0-9][0-9]
Natural Number	[0-9]+
Real Number	[-0-9]+[.,]+[0-9].,]+
Alpha-Num	[A-Za-z0-9]+
Roman	[ivxdlcm]+ or [IVXDLCM]+
Has Dash	.*-.*
Init Dash	-.*
End Dash	.*-
Punctuation	[,.;:?!-+””]

Table 5.2: Orthographic predicates used by our CRF system. The observation list for each token will include a predicate for every regular expression that token t matches

features. For example, comparing the words ‘laser’ and ‘kinase’, it is much more informative to know that ‘ase’ is at the end of a word than just knowing ‘ase’ is in the word.

Dictionary features: We also included two dictionaries, namely a protein - (61676 entries) and a cell type lexicon (204 entries) from Shi and Campagne [2005]. Remember from section 2 that using only a dictionary-based approach usually yields a recall of about 10-30%.

Context Features: Consider the following sentence: ‘The IL-2 gene was localized on mouse chromosome 12’. This example shows that regular expressions alone are not suitable to distinguish between similar entities like gene or protein. According to the orthographic features mentioned before, it wouldn’t be a big problem to assign an entity not being **O** to ‘IL-2’, but which entity should we assign? Thus, it is useful to consider the preceding or following words of a token at a specific position. If the token is ‘IL-2’ and the following word is ‘gene’, then the word ‘gene’ will help the CRF to distinguish between the entities protein or gene. In our approach we use a window size of three to include context features. This means, we look at the preceding, the current and the following word and extract all features listed before for these words. Additionally, we enlarge this feature for some special keywords of the window size five. These special keywords can act as indicators for some entities. They typically occur at the end of an entity. We included the following keywords: gene, protein, mRNA, subunit, receptor or promoter.

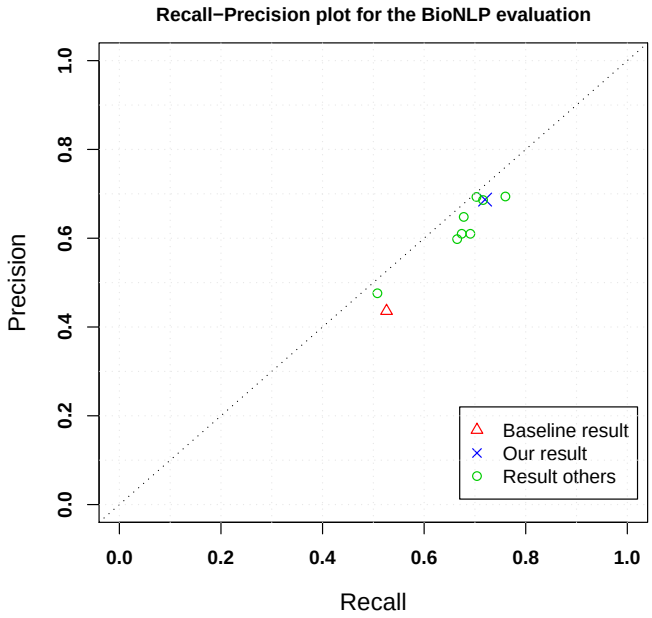

Figure 5.2: Recall-Precision plot for the participants of the BioNLP/JNLPBA shared task. The baseline approach was simply to memorize the entities occurring in the training data and then applying a longest match approach to the test data.

Team	Recall	Precision	F-Score
Zho04	76.0	69.4	72.6
Our result	71.8	68.8	70.6
Fin04	71.6	68.6	70.1
Set04	70.3	69.3	69.8
Son04	67.8	64.8	66.3
Zha04	69.1	61.0	64.8
Roes04	67.4	61.0	64.0
Par04	66.5	59.8	63.0
Lee04	50.8	47.6	49.1
BL	52.6	43.6	47.7

Table 5.3: Results of the BioNLP/JNLPBA shared task. The team names have been adopted from the shared task. The measurements have been averaged over the different entity types (Protein, Rna, Dna, cell line and cell type). Admeasurement as percentage.

Results: Figure 5.2 shows a precision-recall plot for all participants of the shared task. In total we ranked second place, only one system from Zhou and Su [2004] performed clearly better than our approach. Zhou and Su [2004] use an ensemble of classifiers (two discriminative Markov Models and one Support Vector Machine) with a simple majority voting strategy. Additionally, they apply some post-processing methods. We in contrast use only a single general model with no post-processing at all. Finkel et al. [2004] use a MEMM approach and rank scarcely behind us. If there is a statistical significance in the difference, then the difference could be due to the so-called label bias problem with MEMM. One other system also uses CRFs [Settles, 2004] and rank fourth. The main difference to our approach is that they use much more lexicons (about seventeen while we only use two) and they dont't use any n-gram features. In section 5.1.3 we show that using a large amount of different dictionaries is a disadvantage and can lead to a large performance decrease. However, orthographical features were used by almost all participating systems, which stresses the importance in biomedical named entity recognition. In total, five teams used Support Vector Machines alone or in combination with other classifiers. One team used HMMs [Zhao, 2004] alone and ranked sixth. Thus, we clearly outperform the HMM. We achieve our best result with a linear-chain CRF and the features mentioned in the system description. We used the default spherical Gaussian prior to maximize tagging accuracy on the development set. Therefore a single large weight dominating a decision is reduced. When we exclude word shape and dictionary features we achieve an F-Score of 69.58. The F-score result, when including dictionary features to our baseline CRF (including orthographic features, n-gram features and context features) is 70.03 and 70.11 when including only word shape features and no dictionary features. The second-order CRFs all yield worse results than the linear-chain CRFs. These models would probably need much more training data to perform better results than the linear-chain CRF due to their complexity (see figure 5.5).

NE category	Recall	Precision	F-Score
Protein	77.9	68.8	73.1
DNA	63.6	66.9	65.2
RNA	64.4	63.9	64.1
cell type	65.2	78.9	71.4
cell line	57.0	52.5	54.7
All	72.0	69.3	70.6

Table 5.4: Performance of each entity type on the JNLPBA shared task

Table 5.4 list recall, precision and balanced F-score for each type of entity of the shared task. F-Scores for protein and cell type are comparably high. The reason is probably that these entities are among the most frequent entities in the training data (see table 5.1). DNA however, is the second most frequent type of entity in the training data and yields only poor results. This discrepancy can be due to the fact that DNA names are commonly used in proteins, causing a remarkable overlap between those categories. RNA and cell line are less frequent in the training data and therefore cannot compete with entities like protein or cell type. Cell line performance is additionally small, because it strongly overlaps with the category cell type. Remarkable is the high recall for the entity protein (77.7) and the high precision for cell type (78.4). The recall for protein shows that our features match quite well the criteria to identify protein names (low number of false negatives). The low recall for cell type may be due to the overlapping category with cell line.

Sources of Error: Remember from section 1.1.3 that the inter-annotator agreement can be an indicator of the difficulty of the task, i. e. indicating the possible upper limit of system performance. Although there is no inter-annotator agreement results for the GENIA corpus, there is one recent study for biomedical named entity recognition which has measured agreement between 87% [Hirschmann]. One problem for the annotators is the so called preceding adjective problem. It is very difficult for the experts to decide whether descriptive adjectives, such as 'normal', 'human' or 'activated' should be part of the named entities. Another problem is that some type of entities, which are quite similar (e. g. cell line and cell type) are hard to distinguish even for experts. The system recognition errors of our system are:

- **Misclassification**: E. g. some protein molecules or regions are misclassified as DNA molecules or regions.

- **False positives**: Some entities appear without accompanying a specific name, for example, only mention about 'the epitopes' rather than which kind of epitopes. The human experts tend to ignore these entities, but not if they occur with specific names like 'CD4 epitopes'. Our system recognizes the entities without accompanying a specific name as a member of the entity class.

- **False negatives**: Some entities which consist entirely of proper nouns are hard to identify. Our system sometimes misses to tag these entities.
- **Long phrase problem**: Our system has some difficulties with long phrases, consider e. g. the phrase 'TSHR antibody and microsomal antibody'. Our system tends to tag only one entity 'TSHR antibody' instead of tagging both entities.

NE category	Exact Match	Right Match	Left Match	Soft Match
Protein	73.1	80.0	78.2	84.1
DNA	65.2	73.9	67.7	75.9
RNA	64.1	76.8	66.7	80.2
cell type	71.4	80.6	73.0	80.2
cell line	54.7	64.4	57.9	67.3
All	70.3	78.3	74.3	81.2

Table 5.5: F-Scores for different matching criteria for JNLPBA

Results with regard to relation extraction: Until now, we only considered the results with one matching criteria, which we call exact match. This means, that a true positive must match the exact boundaries of the entity. If we missed one boundary, for example the left boundary, we had a false positive, because we identified a wrong entity. In addition, we got one false negative, because we missed to identify the correct entity. Such a hard matching criteria is essential when considering the task with the goal of curating a database. However, if we want to detect the relations between different entities, we can relax this hard matching criteria to softer ones. Table 5.5 shows respectively the F-Scores for the different type of entities for relaxed matching criteria. Right Match means that the right boundary of the entity has to be correct, while Left Match means that the left boundary of the entity has to be detected correctly. Soft Match means, that either Right Match or Left Match has to be detected correctly. With relaxed boundary matching the F-scores increase for the category ALL from 4% (Left Match) to 8% (Right Match) and to 11.9% for Soft Match. In general, the degree of relaxing the matching criteria strongly depends on the type of bottom-up application one wants to use [Tsai et al., 2006] .

5.1.2 Critical Assessment of Information Extraction Systems in Biology - BioCreAtIvE task 1A

In general, the goal of the BioCreAtIvE challenge is to provide a set of common evaluation tasks to assess the state of the art for text mining applied to biological problems. One task at this conference was the extraction of gene or protein names from text (task 1A) and their mapping into standardized gene identifiers for three model organism databases (fly, mouse and yeast)(task 1B) from PubMed abstracts. In contrast to the BioNLP/JNLPBA

evaluation described before, the task was only to identify the entity 'gene or protein' (Usually this entity consists of two types of entities, namely gene and protein, however in the BioCreAtIvE challenge 1A they were merged together). The entities which are marked as gene or protein mention consist of genes, including binding sites, motifs, domains, proteins and promoters. Yeh et al. [2005] mentions as difficulties of the task, that gene or protein mentions are often English common nouns. They are often descriptions and they often include ordinary words like 'blistery','inflated', 'period', to mention a few examples from Drosophila gene names. The training data consists of 7500 sentences plus 2500 development test sentences. The (final) test set consists of 5000 sentences. The development set can be used for parameter optimization or can also be used as additional training data. Half of the sentences from Medline abstracts were chosen from abstracts likely to contain gene or protein names and the other half of the sentences were chosen not to contain such names (see Tanabe et al. [2005] for details on the construction of the data for task 1A). The scoring of the BioCreAtIvE task 1A differs from the scoring of the JNLPBA evaluation in the following way: consider again following sentence:
' ... a protein related to ***SNF1* protein kinase**'.
The italic boldface indicates the allowed alternatives for the entity 'SNF1 protein kinase'. This means that in order to get a TP, one needs to match the entity itself or one of its alternatives (in this case 'SNF1'). The rules for getting false positives and false negatives remain the same as in section 5.1.1. The BioCreAtIvE task 1A had two possible submission forms: open and closed form. Open form means, that the teams incorporate additional information to the training data (e. g. dictionaries). In general, open submissions yielded better results than closed submissions. An exception was team B (see table 5.7). Note that the BioCreAtIvE team aimed at making the task of identifying entities very realistic to a real world application. Thus, they also included only sentence fragments in the training and test data to make the task more difficult. Sentence fragments occur in the real world, because on the one hand it can be a mistake of the author and on the other hand, no perfect sentence segmenting algorithm exists right now. Preprocessing of the data was provided (tokenization and POS-tagging).

Data Set	Sentences	Gene Mentions
Training Set	7500	9000
Development Set	2500	3000
Final Test Set	5000	6000

Table 5.6: Number of sentences and number of gene mentions for the different data sets at BioCreAtIvE.

System Description: Our conditional random field for the BioCreAtIvE task 1A looks exactly the same as the CRF from the JNLPBA evaluation task. The features and the parameters of the model are the same. The only markable difference is, that since we are not searching for cell type entities the correspondent dictionary serves as composite feature.

Results: Figure 5.3 shows the recall-precision plot for the BioCreAtIvE evaluation task 1A open submission. In total we ranked third, only one team performs clearly better than us and the second team performs slightly better. Only four teams (including our result) managed to break the F-Score boundary of 80.0. The first team is the same team as in the JNLPBA shared task which uses the ensemble of classifiers with the extensive post-processing [Zhou et al., 2005]. Without post-processing their simple majority voting strategy achieves an F-score of 76.4. So their post-processing methods yields a performance improvement of 6.8 %. For this purpose, they use methods like abbreviation resolution and name refinement. In the future we should also consider to incorporate some post-processing methods to improve performance. The second team also uses a conditional random field approach [Mcdonald and Pereira, 2005]. The better results are probably due to larger dictionaries and the use of a feature induction algorithm [Mccallum and Li, 2003]. This method tries to find feature conjunctions which would significantly improve the log-likelihood when added to the model. When we are using the feature induction algorithm of McCallum, our results show no improvement. Interestingly, the two CRF applications have the highest precision rates (86.4% and 84.2%). This time our dictionary feature gives much more performance improvement when using it (approx. 3% F-measure improvement vs. 0.5% improvement at JNLPBA). At JNLPBA we only provide two dictionaries and have to predict five entities, while here we also provide two dictionaries but have only to predict one single entity.

In general the participating systems either used some type of Markov modeling, a SVM or a rule-based approach. The systems combined these approaches with some pre- and/or post-processing stages. Many systems also used a part-of-speech (POS) tagger to find a word‘s part-of-speech and incorporated them as features in their system.
Many of the high performing systems achieved scores quite close together. They were close enough to be affected by the disagreements in annotation or to be not statistically significant. At a normal threshold of 5% for statistical significance team A and team C were borderline significant according to some randomization tests accomplished by the BioCreAtIvE evaluation.

Sources of Error: The sources of error were the same like from the JNLPBA evaluation task, except the misclassification errors which could not happen, since the task was only to identify one entity.

Results with Regard to Relation Extraction: Again we want to consider the F-scores under softer matching criteria. Table 5.8 shows the F-scores for different criteria. Right Match yields a performance improvement of 3.5%, Left Match 3.8% and Soft Match an improvement of 7.3%. These improvements are significantly lower than the improvements for the different matching criteria for the JNLPBA shared task. The reason is, that the BioCreAtIvE evaluation also allowed alternative TPs which already often matched some softer criteria for the identification of the entities.

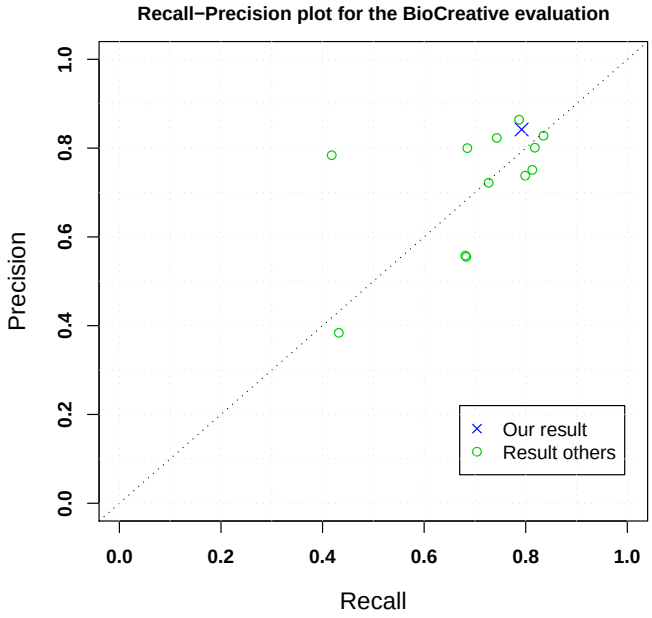

Figure 5.3: Recall-Precision plot for the participants of the BioCreAtIvE shared task 1A.

Team	Recall	Precision	F-Score
A	83.5	82.8	83.2
C	78.7	86.4	82.4
Our Result	79.2	84.2	81.6
D	81.8	80.1	80.9
B	81.3	75.1	78.1
E	74.3	82.3	78.1
G	79.9	73.8	76.7
H	68.5	80.0	73.8
I	72.7	72.2	72.4
J	68.1	55.8	61.3
K	68.3	55.5	61.2
M	41.8	78.4	54.5
O	43.2	38.4	40.7

Table 5.7: Results of the BioCreAtIvE shared task 1A open form. The team names have been adopted from the shared task. Admeasurement as percentage.

NE category	Exact Match	Right Match	Left Match	Soft Match
Protein	81.6	85.1	85.4	88.9

Table 5.8: F-Scores for different matching criteria for BioCreAtIvE task 1A

5.1.3 Language-Independent Named Entity Recognition at CoNLL-2003

The shared task of CoNLL-2003 concerns language-independent named entity recognition of persons, locations, companies and miscellaneous entities. The objective of the shared task was to set up a named-entity recognition system that includes a machine learning component. The shared task organizers were especially interested in approaches that made use of external knowledge (knowledge not supplied by the training data) like gazetters or unannotated data. The challenge was to incorporate the unannotated data in one way. Since we evaluate only the German data set for this conference, we omit the details for the English data. Training data consists of 12705 sentences and 3068 sentences for the development data. The final test data has 3160 test instances. Table 5.9 summarizes the distribution of the entities for the different sets. In the training set the person entities (PER) occurs most often after the miscellaneous entities (MISC) and the location entities (LOC). The miscellaneous group is something special for this named-entity task, other evaluation conferences for named-entity recognition do not provide such a group. Examples for entities belonging to this group in the German data set are historical events like the 'Second World War' or the 'November Revolution' or some specific awards like the 'Grammy Award' or names of songs, e. g. .

For constructing the different data sets, the CoNLL-03 organizers applied the tokenization, part-of-speech tagging and text chunking. The German data was lemmatized, tagged and chunked by the decision-tree Treetagger[1]. The POS-tags and the text chunks are also part of the training data and were used by almost all participating systems. For the test data set, these features were not provided. The systems had to use their own taggers. Again, we did not use any of this kind of deeper syntactic knowledge, since we wanted to put weight on the performance of our application.

German Data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	553	12705	206931	4363	2288	2427	2773
Development set	201	3068	51444	1181	1010	1241	1401
Test set	155	3160	51943	1035	670	773	1195

Table 5.9: Statistics for the different German data sets of the CoNLL-03 shared task

[1]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

System Description: We used the same type of features for this kind of NER task (see section 5.1.1). There were only two minor changes with respect to the CRF models of the bio-entity recognition tasks. First, we excluded the so-called Word Shape Feature, since these reduced system performance (0.7%). Second, we use other dictionaries for this task. We collect dozens of dictionaries from web resources, and unify them in three dictionaries (person lexicon, location lexicon and an organization lexicon). Additionally, we use the lexicons provided by the CoNLL03 shared task organizers and incorporate them in the three lexicons mentioned before.

Team	Recall	Precision	F-Score
Our result (a)	69.75	85.53	76.84
Florian	63.71	83.87	72.41
Klein	65.04	80.38	71.90
Our result (b)	61.85	85.50	71.78
Zhang	63.03	82.00	71.27
Mayfield	64.82	75.97	69.96
Carreras (b)	63.82	75.47	69.15
Bender	63.82	74.82	68.88
Curran	62.46	75.61	68.41
McCallum	61.72	75.97	68.11
Munro	66.21	69.37	67.75
Carreras (a)	58.02	77.83	66.48
Wu	59.35	75.20	66.34
Chieu	57.34	76.83	65.67
Hendrickx	56.55	71.15	63.02
De Meulder	51.86	63.93	57.27
Whitelaw	44.11	71.05	54.43
Hammerton	38.25	63.49	47.74
baseline	28.89	31.86	30.3

Table 5.10: Results of the CoNLL-03 task. Our result (a) has used the development data as additional resource of training data, while Our result (b) has simply used the conventional training data. The baseline results have been produced by a system which only selects complete unambiguous named entities which appear in the training data. The names of the team have been adapted from the shared task. Admeasurement as percentage

Results: Figure 5.4 shows the recall-precision plot for the CoNLL-03 named entity task. We present here two different results. The first result (called Our Result (a) here) includes some additional training data, namely the development data provided by the CoNLL-03 organizers. This data is originally thought for the parameter tuning of the participants, but since we don‘t perform any parameter tuning (we use the standard spherical gaussian prior of 1.0), we just use this data as additional training data. Our second result (called

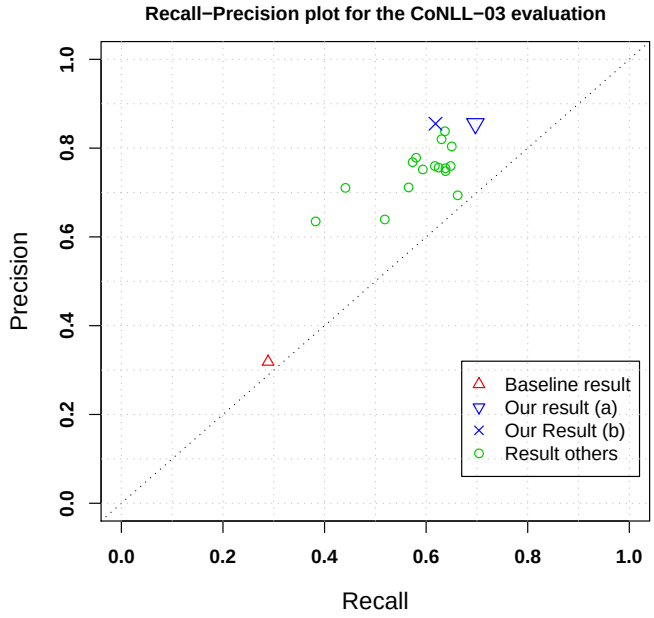

Figure 5.4: Recall-Precision plot for the participants of the CoNLL-03 named entity recognition task. Our result (a) included additional training data, while Our result (b) contained the original training data

Our Result (b) here) is achieved by a model which was trained with the original training data. The use of additional training data (3068 additional sentences) results in an F-measure increase of approx. 5%. Surprisingly the precision rate for the two models is almost the same (about 85%), while the most improvement can be achieved through the better recall rate (about 8% difference). Like in the other evaluations, the CRF models achieve the highest precision rates compared to all other systems. Second order CRFs and the attempt to reduce the feature space with the help of the feature induction algorithm [Mccallum and Li, 2003] all yielded worse performance.

The most frequently applied technique in this named entity task was the Maximum Entropy Model. In total five systems used this approach, three of them used this model in isolation and two systems used this technique in combination with other algorithms. Hidden Markov Models were employed by four systems, but they were always used in combination with other techniques. In contrast to the bio-entity tasks, only one team used support vector machines. Conditional Random Fields were used only by one team [Mccallum and Li, 2003]. One team used a kind of meta named entity recognizer by combining their systems with other existing external NER systems [Florian et al., 2003].

McCallum‘s systems achieves a performance of 68.11 which lies clearly behind our result. McCallum argues that they only use five lexicons for the German NER task. In what follows, we compare his system with Our Result (b) for comparability reasons. We

use three large lexica, and surprisingly in this task the lexica give no performance improvement in our case. The model without dictionary features matches the result of the model with dictionary features (0.02% difference). Astonishingly,when using no lexica the performance for tagging locations and organizations increases (1% and 2% improvement), while the performance for persons decreases (2% decrease). McCallum does not provide complete details about their system configuration, but we assume, that we use other orthographic features and additionally we are using prefix - and suffix features, while he is using only character bi- and tri-grams. Another reason for the performance difference lies perhaps in the use of several small lexica instead of using less but larger lexica (as in our case). McCallum, uses a lexica for universities and another one for sport clubs, e. g. . We merge all these different organizations into one large lexicon instead. If in the training data universities occur quite rate, the weight for the university lexicon feature will be considerably low. Now assume that university names occur quite often in the test data. As a consequence, the corresponding feature in McCallum‘s configuration will be low and therefore his system has to rely on other features at this point and false negatives/false positives can occur more easily.

All teams used a variety of features. Only one team did not use any lexical features. Most of the systems employed part-of-speech tags and chunking features. However our result suggests, that it is absolutely possible to achieve state of the are performance without using this kind of syntactic features. Orthographic and affix features were also incorporated by most of the systems. Erik et al. [2003] provide an overview table for the main features used by the participating teams.

Table 5.11 compares our two results with the results of the best participating systems [Florian et al., 2003]. Remember that this team used existing external NER systems and combined them with their named entity recognizer. When comparing Our Result (a) with Our Result (b), it strikes that the precision values for the two results are already almost the same. So, when adding additional training data, an increase of the performance can only be achieved in terms of recall. This especially holds for the person tagging, where we can achieve an approx. 12% recall improvement. But also for location and organization tagging, a significant approx. 5% recall improvement can be achieved. Interestingly, the first three teams outperform our person result. Our strengths lie clearly in tagging locations and organizations which make up about 60% of all entities (see table 5.9).

	Our Result(a)			Our Result(b)			Florian		
	Recall	Precision	F-Score	Recall	Precision	F-Score	Recall	Precision	F-Score
PER	72.81	89.90	80.46	60.41	90.69	72.52	75.31	91.93	82.80
LOC	74.79	87.72	80.74	69.65	88.41	77.91	71.59	80.19	75.65
ORG	65.41	74.61	69.71	60.96	73.86	66.79	54.46	79.43	64.62
MISC	38.35	79.00	51.63	33.01	74.73	45.79	41.49	77.87	54.14

Table 5.11: Comparison of our two results with the best participating system for the CoNLL-03 evaluation

Sources of Error: We also compare again softer matching criteria (see Table 5.12). Here, the results do not increase as drastically as in the case of identifying biomedical entities. We get an overall performance improvement of about 3,5% for the soft match criterion, while in the JNLPBA task we get an improvement of over 12% (see table 5.5) and for BioCreAtIvE an overall improvement of about 7% (see table 5.8). The reason for more boundary errors in the task of identifying biomedical entities could be due to the reason that the biomedical entities are much more similar to each other, which can simply lead to misclassifications (see paragraph Sources of Error in section 5.1.1). Another reason could be that in biomedical entity recognition the distribution over the length of the entities is different than in the case of the traditional NER task. The entities tend to be longer in the case of bio-entities, which makes the task more difficult. Third, there is no committing convention for assigning names for gene or proteins. Besides the common errors like false positives and false negatives other sources of errors can be:

- **Misclassification:** Our system has clearly problems with entities which are not typical for a specific entity like 'Dreikönigskeller' or 'Irish Pub' for example. This is not a typical location and can easily lead to misclassification. Another example are organizations, when their name contains a persons name and are referenced in the text without any additional knowledge. E. g. if a companies name is 'Holzmann Gmbh' and this company is mentioned in the text only with 'Holzmann' our system tends to recognize this as a persons name. When the context of the sentence is missing, even human annotators will have labeling problems.
- **Phrases with lots of entities:** If a phrase contains a lot of different entities our systems mostly merges this phrase to one large entity instead of making more entities. Consider the phrase 'Nina Hagen Ovationen für Brecht'. Since our system does not use any syntactical deeper knowledge, it is confused about the large amount of words with capital letters and therefore tags the whole phrase as one entity instead of tagging it as several entities.

NE category	Exact Match	Right Match	Left Match	Soft Match
PER	80.46	82.31	81.76	83.07
LOC	80.74	83.59	80.69	84.07
ORG	69.71	72.23	73.14	75.22
MISC	51.63	51.63	51.63	51.63
ALL	76.84	79.10	78.04	80.04

Table 5.12: F-Scores for different matching criteria for the CoNLL-03 task (Our result (a))

5.1.4 Named Entity Recognition at MUC-6

The Message Understanding Conference (MUC-6) was hold in October 1995. The conference provided several evaluation tasks, where one of them was a named entity (NE)

task. The text is annotated with the Standard Generalized Markup Language (SGML) and consists of the following tag elements:

- ENAMEX (for entity names, consisting of organizations, persons and locations)
- TIMEX (for temporal expressions like dates and times)
- NUMEX (for number expressions, i. e. currency values and percentages)

These element types have attributes, namely for ENAMEX the attribute types can be ORGANIZATION, PERSON or LOCATION. For TIMEX elements the attributes can be either DATE or TIME. For NUMEX elements the attributes are MONEY or PERCENT. The corpus was made of Wall Street Journal articles and was drawn from a corpus of approximately 58000 articles spanning a period from January 1993 through June 1994. The training set consists of 100 articles and the test set is 30 articles large. Table 5.13 gives an overview over the entities for the training and test data. The training and test data was tokenized, but no POS-tags or text chunks were provided for the training (for details on the tokenization, see Grishman and Sundheim [1995]).

English Data	Articles	Sentences	LOC	ORG	PER
Training Data	100	6773	2319	3730	2059
Test Data	30	543	87	463	373

Table 5.13: Statistics for the different English data sets of the MUC-6 NE task

We did not evaluate our approach on the TIMEX and NUMEX entity tags. The results of the MUC-6 conference on these entity tags indicate, that these element tags can be extracted nearly perfectly (half of the systems did not make any mistakes) by applying simple pattern matching algorithms. Therefore we simply skipped these entity tags, in order to be capable to compare the named entity task for English in respect to German. Thus, we cannot directly compare the final results of the participating teams of MUC-6 with our result. Additionally, there were some optional ENAMEX entities in the test set (44 those entities), which when missing or tagged wrong did not count as a mistake for the systems. Since there were no such optional entities in the CoNLL-03 shared task, we assigned a mistake when we missed such an optional ENAMEX entity.

The results for the original setting of the MUC-6 conference are in general very good. Fifteen teams participated in the task and more then half of the teams achieved an F-Score of over 90%. One system even matched human performance (96.68%, also referred to as inter-annotator agreement in former sections). If we consider these results from a general point of view some facts must be stated. First, the test data is very small (only 30 articles). Second the test and training set represents a certain style of documents and writing. The style of writing is clearly journalistic and reports about financial news. Therefore, it can not be expected, that this excellent result also holds in general for the task of identifying persons locations and organizations. This would also coincide with the results of the

CoNLL-03 shared task for English, where the best team stated an F-measure of 88.76. 80% of the MUC-6 test data are ENAMEX types, the remaining 20% are TIMEX and NUMEX entities. As already mentioned, TIMEX and NUMEX entities can be predicted almost perfectly. This has additionally a clearly positive effect for the overall F-measure. The test set has in total 863 ENAMEX entities and 173 TIMEX and NUMEX entities. If we assume, that we are able to tag TIMEX and NUMEX nearly perfectly, than we would have additional 173 True Positives.

System Description: We skipped the character bi- and trigrams for this evaluation, since they decreased performance on the test set. In other respects our features for the CRF model did not change from the CoNLL03 CRF model. Again, we used only linear chain CRFs and the feature induction algorithm did not show improvement.

NE category	Exact Match	Right Match	Left Match	Soft Match
PER	93.24	93.53	93.24	93.79
LOC	87.43	89.62	87.43	90.01
ORG	81.67	85.36	84.13	86.33
ALL	87.05	89.19	88.24	89.98

Table 5.14: F-Scores for different matching criteria for the MUC-6 task

Results: Table 5.14 shows the F-Scores for the different entities under different matching criteria. As already mentioned, we cannot directly compare the results with the MUC-6 participants. But if we look at the best participating system, it becomes clear, that we loose almost all performance in comparison to the best system, when trying to tag organizations. We have only an F-Measure of about 87% in comparison to the best team which has an F-Measure for organizations of about 93%. 33 of the 44 optional ENAMEX entities were organization entities, so we would likely score better with the MUC-6 evaluation software, but we would not reach the performance for organizations of the best participating team. Additionally, about half of the ENAMEX entities of the test set have the attribute ORGANIZATION, so the performance on this type has a large impact on the overall performance. In the case of tagging the person entities we seem to keep up with the best participating systems. Our score for locations seems to coincide quite good with the result of the best participating system (87.4% in comparison to 89%), but there are unfortunately only 87 samples in the test data, so these results are not really representative. In addition, we compared the use of different dictionary settings. We first tried the dictionary setting from Mccallum and Li [2003] from CoNLL-03 (many but smaller dictionaries) and afterwards tried our dictionary setting from CoNLL-03 (larger but less dictionaries). We can observe a 3% F-measure increase, when using our dictionary setting.

When considering softer matching criteria, it strikes that we get the most performance improvement in the case of tagging organizations (about 5% improvement for the soft

match criterion). For the other type of entities we only get slight improvement or nearly no improvement. We get an overall improvement of about 2%, that is even less than in the CoNLL03 task evaluation, where we got about 4% improvement.

Sources of Error: Again, we have much less boundary errors in this kind of NER task than in the biomedical domain. This does not hold for the ENAMEX entity with the attribute ORGANIZATION (see table 5.14). If an organization is mentioned in a sentence with a preceding 'the', our system tends to tag this, therefore a boundary error occurs. Additionally, if an organization stands at the beginning of a sentence and there is only one preceding word (which is of course written in capital letters), our systems tags the first word as part of the entity. Other sources of error are:

- **False Positives:** Technical terms like M. B. A or SX (which refers to a special kind of computer workstation) are mostly tagged with an organization tag. In phrases like 'the Solaris software' the MUC-6 annotators tend to discard Solaris as an organization, since it is somehow mentioned in another context, however, our systems also tags these cases.

- **False Negatives:** Entities, especially organization entities, which are composed of normal words like a company named 'Birds Eye' are not recognized by our system, if they are not in the dictionary. If entities begin with a common word, 'All-American Gourmet Co.' , e. g. , our system produces a boundary error, since it does not recognize the first common word as part of the entity.

- **Long Phrase Entities:** Our system tends to tag several entities instead of one entity, if the phrase to be tagged is very long. This especially holds for lawyers agencies: 'Proskauer, Rose, Goetz & Mendelssohn' are tagged as one entity by the human annotators, our system tags each single partner as a single organization.

- **Heading problems:** Headings from the Wall Street Journal pose a challenge for our CRF model, since the words occurring in the headings are all in capital letters. The word shape feature is however an important feature for tagging normal text and thus has largely positive weight in our model. So we get a lot of false positives for headings. If we delete the headings from the test data, we get an performance improvement of about 2%.

5.1.5 Summary

We have shown that CRFs are a suitable probabilistic model for named entity recognition for various domains and different languages. Note that the CRF implementation is a single probabilistic tagging model. This model is quite general, since there is no domain-specific pre- or postprocessing. It can be easily applied to various other biological entities and even other domains, provided appropriate lexicons are available. In all the evaluations the

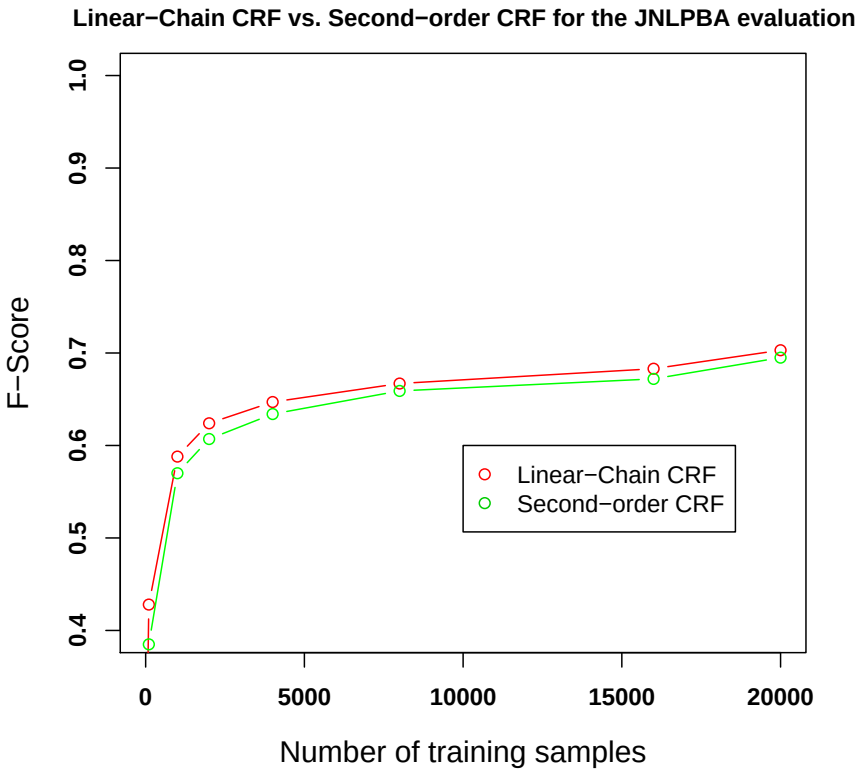

Figure 5.5: Linear-chain CRFs compared to second-order CRFs. Here, the number of training samples vs. F-measure is plotted

CRFs showed very competitive performance, while using the same type of features. We are able to tag unseen sentences with high performance, because we pass on using any deeper linguistic features or deeper syntactic knowledge. Recall from section 1.1.1 that we consider deeper syntactic knowledge to be everything which goes beyond tokenization. We conclude that in order to achieve state of the art performance in NER, we can do without linguistic features. Throughout the evaluations we did not optimize the default spherical prior. Thus, we could expect some further performance improvement, when adjusting that parameter. See 6.2.1 for further ideas on improving performance on NER. As could be seen in the CoNLL-03 and in the MUC6 evaluation, the possible performance is highly dependent on the quality and representativeness of the training corpus. CRFs can get confused about different styles of documents (see heading problems in section 5.1.4). English NER is a much easier task than german NER, because of the lower complexity of the language. In biomedical named entity recognition, orthographic features had the highest positive weights, while in the newswire domain word shape features were dominant. Thus, the importance of different features varies with the domain. The use of dictionary features could improve performance significantly in two evaluations (BioCreAtIvE and MUC-6), while in the other evaluations performance did not increase significantly or even not at all. Thus, the importance of dictionaries is dependent on the data set. Furthermore, it can be concluded that well chosen dictionary-based features can significantly improve results on unseen data (see section 5.1.4). Boundary errors are much more apparent in the biomedical domain. In general, biomedical named entities tend to be longer and thus making it more difficult to recognize the boundaries properly.

Throughout all experiments feature induction could not boost performance. In our setting, linear-chain CRFs were always superior to the more complex second-order CRFs (see figure 5.5). Second-order CRFs probably need more training data, in order to outperform linear-chain CRFs, because of their higher complexity.

Training time is quite expensive even for linear-chain CRFs and is dependent on the number of entities and training sample size. Training took about 15h in the BioNLP/JNLPBA evaluation (five types of entities and about 20000 training sentences) for a linear-chain CRF, while training time for the BioCreAtIvE evaluation took about 4h for 7500 sentences and one type of entity. Once training is done, assigning new entities to unseen examples is pretty fast (approx. 0.2 sec for tagging one sentence). All the experiments were conducted on a common workstation (Genuine Intel(R) CPU T2300 @ 1.66GHz 980MHz, 0.99 GB RAM).

5.2 Relation Extraction

In this section we describe the results of extracting protein interactions from text. The problem considered here is that of identifying interactions between proteins from biomedical literature. As already described earlier, there is unfortunately no existing standard benchmark data set for extracting protein interactions from text. Nevertheless, there are some corpora available on the web. The Interaction Extraction Performance Assessment

(IEPA) corpus [Ding et al., 2002] is a body of 303 Medline abstracts using ten different queries. Each query consists of two proteins and was combined through a logical AND. Suggested queries were chosen only if the number of abstracts retrieved exceeded ten. The AImed corpus [Bunescu et al., 2005] consists of 230 abstracts, two hundred of them were obtained from the Database of Interacting Proteins (DIP) [Xenarios et al., 2000] and were previously known to describe interactions. These two hundred abstracts contain 1101 interactions and about 4141 protein mentions. Additionally, 30 abstracts were added, since only a few examples mentioned exactly two proteins not interacting with each other. The Learning Language in Logic (LLL) challenge on Genic Interaction extraction [Nedellec, 2005] is the only data set, which provides an independent test set, but training - (80 sentences) and test data (87 sentences) are too small in order to be applicable for machine learning algorithms. Since the IEPA corpus is constructed using only ten queries, we expect that the proteins involved in interactions are very biased and therefore it could be very easy to learn interactions on this data set, when using some information from protein entities. As a consequence the AImed corpus seems to be the most realistic and most diverse data set for the relation detection task. In addition, some recent publications [Bunescu et al., 2005, Bunescu and Mooney, 2005] conducted experiments on this data.

Zhou et al. [2005] did some experiments on the ACE RDC task (see Section 3) and report a 11 F-measure improvement on relation detection and a 20 F-measure improvement on relation detection and characterization over tree kernel-based approaches. They incorporate diverse lexical, syntactic and semantic knowledge for feature-based relation extraction and applied a SVM. Inspired by this work, we also want to test feature-based relation extraction with a SVM on the AImed corpus for relation detection. Furthermore, Zhou et al. [2005] performed about 20% F-measure better than Bunescu and Mooney [2005] on the ACE RDC task. In addition, we want to incorporate ontology features from Gene Ontology (GO) [Harris et al., 2004] and other external features from existing protein databases. To the best of our knowledge, the idea of using ontology information or other external knowledge for extracting relations from text is new and hasn‘t been published in the NLP community yet.

Recently, ontologies have gained a lot of patience in the AI community. Ontologies can be described as specification of a conceptualization. Thus, an ontology is a description of the concepts and relationships which can hold for an agent or a community of agents. One main purpose of ontologies is knowledge sharing and reuse of knowledge. The GO project addresses the need for consistent descriptions of gene products in different databases. GO has developed three controlled vocabularies that describe gene products in terms of their cellular component, molecular functions and their associated biological process. With the help of GO, it is very easy to query all gene products in the mouse genome that participate in signal transduction, e. g. . The structure of GO also allows annotators to assign properties to genes at different levels, dependent on the depth of knowledge about that gene. The gene products in GO implement two relationships: *is_a* and *part_of*. E. g. the GO term **regulation of cell cycle** and the term **regulation of cellular physiological process** encode a *is_a* relationship. If a gene has the term **regulation of cell cycle** it is automatically *part_of* **regulation of cellular physiological process**. Thus, the gene

(1) **BMP-2** antagonists emerge from alterations in the low - affinity binding epitope for receptor **BMPR-II**.

(2) **Bone morphogenetic protein-2** (**BMP-2**) induces bone formation and regeneration in adult vertebrates and regulates important developmental processes in all animals.

(3) **BMP – 2** [1,1] is a homodimeric cysteine knot protein that , as a member of the **transforming growth factor-beta**(**TGF - beta**) superfamily , signals by oligomerizing type I and type II receptor serine-kinases in the cell membrane.

(4) The binding epitopes of **BMP – 2**[1,2,3] for **BMPR – IA**[1] (type I) and **BMPR-II**[2] or **ActR-II**[3] (type II) were characterized using **BMP-2** mutant proteins for analysis of interactions with receptor ectodomains.

(5) The extracellular domain of the human **neurotrophin TRKB receptor**[4,5] expressed in Chinese hamster ovary cells is a highly glycosylated protein , possessing binding ability for **brain-derived neurotrophic factor**[4] (**BDNF**[5]).

Figure 5.6: Example text phrases of the AImed corpus with all proteins and interactions tagged. Protein names have been highlighted and the numbers in brackets indicate interactions between proteins.

ontology is represented as a Directed Acyclic Graph (DAG).

Furthermore, in our feature-based relation extraction system we want to make use of an idea recently described by Xu et al. [2006]. This work aims at detecting relations between customers and products and suggests to make use of the different attributes provided by the various entities. Section 5.2.1 describes the various features in detail and their motivation of using them. This section also describes how we implement the idea of Xu et al. [2006] and describe implementation details. Section 5.2.2 shows the results of our approach on the AImed corpus.

5.2.1 System Description

Figure 5.6 shows some example text phrases from the AImed corpus. Here, an instance is a pair of proteins. If a sentence has n entities, $\binom{n}{2}$ instances are created. If the extracted instance is known to represent an interaction, then the instance is added to the set of corresponding positive examples, otherwise it is added to the set of negative examples. The relations in this data set are assumed to be symmetric. Consider the fourth sentence from figure 5.6: From this sentence ten instances are extracted, three positive examples indicating some interaction and seven negative examples indicating that no interaction is stated for these instances in this sentence. However, sentence three demonstrates one exception for the extraction of instances. If proteins are self-interacting, e. g. a protein

is forming a homodimer complex [2], then we extract one positive instance for each self-interaction (BMP-2 interacting with BMP-2) and one negative instance for each protein pair not interacting (BMP-2 not interacting either with transforming growth factor-beta nor with TGF-beta). Note that despite the fact that transforming growth factor-beta and TGF-beta represent the same protein, we extract two negative examples, since our system has no abbreviation resolution component. Note that we treat the problem of relation detection here as a kind of text classification problem. The basic text unit is a sentence and the task is to assign to the sentence, whether the sentence belongs to the class describing an interaction or not. Thus, our approach can be seen as a kind of bag-of-words approach, with the difference that we don't exploit term frequencies.

Our system can represent a relation between proteins in two different ways. On the one hand, a relation can be described by the entities involved in the relation. On the other hand, a relation can be described by the context the two entities are embedded. Thus, we can distinguish between protein - and interaction features. In this context a protein feature is a feature which represents some properties of a certain protein (i. e. one entity) and an interaction feature represents some properties of a given protein pair (i. e. a pair of entities). In section 5.2.1 we describe, how these different kind of features are incorporated into our system. In what follows, we first list the set of protein features and second, we describe the set of interaction features used in our system. As already mentioned earlier, we make use of external features. External features[3] are properties of proteins or interactions which are extracted with the help of additional knowledge bases (e. g. databases or ontologies). Non-external features are directly extracted from the given sentence in which the instance occurs.

Protein features:

- **Protein name tokens:** We extract the tokens of each protein name. Many protein names already consist only of one token, however, many do not. We remove uninformative protein tokens like stop words, digits, common words like 'protein' or special characters like dashes or brackets inside a protein name. *MMP-12* would result in one remaining protein name token, namely MMP, since all other tokens are removed (see section 1.1.1 for more details on tokenization).

- **Official gene symbol*:** Since it is not common that authors use the official gene symbol of a protein and since a protein usually has several synonyms, we extract the official gene symbol for a given entity. This aims to reduce the data sparseness problem for protein features. The AImed abstracts consists of abstracts discussing human protein interactions, therefore we use the HUGO Gene Nomenclature Committee (HGNC) [White et al., 1997] mapping file. If there is no official gene name for a given entity, we skip this feature.

[2] A dimer is a molecule consisting of two subunits (i. e. two monomers). 'Homo' is indicating, that the two monomers are identical

[3] External features will be highlighted with *

- **Head of protein:** The head of protein is referred to a token indicating the role of the protein. We have constructed a keyword list, that contains names, indicating such a function (e. g. , kinase, receptor, collagenase, activator and inhibitor). If a protein does not contain such a keyword we try to extract some important tokens with the help of suffixes. We assume that very often such indicator words end with certain suffixes like '-ase' or '-in'. If no useful head can be extracted, we skip this feature. This feature shall exploit the fact that receptors are more likely to interact with hormones or other ligands.

- **GO id*:** For each protein we extract all available GO id's if available and prefix it with their corresponding concept, i. e. either molecular function or biological process or cellular component. This feature also aims to reduce the data sparseness problem for proteins. If no GO code is available for this feature, we skip it.

- **Molecular class of protein*:** This feature is extracted from the Human Protein Reference Database (HPRD) [Peri et al., 2003] and reflects a kind of classification of the protein. Examples of molecular class from HPRD are 'Cytokine', 'RNA binding protein', 'Transport protein' or 'Cell Cycle Control protein'. Again, data sparseness shall be reduced. In addition, the fact that certain molecular classes are more likely to interact with other molecular classes shall be exploited.

Interaction features:

- **Tokens between:** We extract the tokens between a protein pair and treat each token as a feature. The tokens between a protein pair are discarded, if tokens are part of another entity and if the entity is located between the protein pair. This feature is used, because the words between a protein pair are often responsible for describing a relation between them.

- **n tokens before:** We use n tokens before the protein pair as feature. Again, if a protein entity is in the n tokens, the tokens of the protein entity are discarded. It's clear, that a lot of relations are described with the help of the tokens before the protein pair.

- **n tokens after:** n tokens after the protein pair are also extracted, since it is obvious that interactions are often described with the tokens after the protein pair. Entity tokens inside the n tokens are again discarded.

- **Keywords:** Temkin and Gilder [2003] provide a keyword list with stemmed verbs indicating an interaction. These verbs are classified into 19 categories (e. g. Break Bond, Inactivate, Cause). Each category contains several verbs indicating a certain interaction. Once again, we distinguish three different positions, where the keywords can appear: (1) between a protein pair; (2) within n tokens before a protein pair (3) within n tokens after a protein pair. We assume that if a keyword occurs in the

mentioned positions, it should be very likely that there is an interaction between the given protein pair.

- **Overlap:** This feature counts the number of entities between a given protein pair. We assume that if the number of entities between a protein pair is large, it should be less likely that these two entities are describing a relation.

- **Number of tokens between:** If the number of tokens between a pair is small, we assume that it is more likely that they interact with each other.

- **GO level*:** von Mering et al. [2002] measure the accuracy of recently predicted protein interactions with help of the GO level two interacting proteins are sharing. They conclude that if two proteins share a common level in the GO graph which is reasonable deep, they are much more likely to interact with each other. We identify the first common node of two proteins in the DAG of GO and extract the distance to the root. We do that for each controlled vocabulary, i. e. molecular function, cellular component and biological process. If one protein of a pair does not have a gene ontology, we assign the root as common level sharing.

As in the NER task from section 5.1, we pass on using any deeper linguistic features. All the experiments were performed using the SVM *LIBSVM*[4] package. In addition, we add cross-validation functionality. Originally, cross-validation is a method for estimating the generalization error based on resampling of the data. Cross-validation is very often used for choosing among several models. The models vary in their parameter setting or in the use of different features. However, when performing parameter and/or feature selection and no independent test set is provided, the performance of the best model is a biased estimate of the true performance and an 'external validation' has to be added to the process of building a model from the training data [Zhang et al., 2006]. The most popular kinds of cross-validation are K-fold Cross-Validation and Leave-One-Out Cross-Validation (LOOCV). In k-fold cross-validation the original data set is splitted into k equal sized data subsets. $k-1$ subsets are used to train the model and the remaining single subset is used for testing the model. This process is repeated k times with each of the subsets exactly used once as validation data. The k results can be averaged and be used to estimate a single performance estimation. LOOCV uses only one instance as validation set, the remaining instances are used as training set. This process is repeated until each observation has been used once as validation data. Remember from section 4.2 that in the SVM approach there are basically two parameters to tune: the kernel function and the cost parameter C. We skip to tuning these parameters and only apply the most simple linear kernel and use the default parameter $C = 1$ for the cost factor, even though other kernels can perform better. We validate our results using 10-fold cross-validation.

As already mentioned, we distinguish between protein - and interaction features. Using this point of view we can produce different kernels for describing protein interactions. In

[4]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

section 4.2 we introduced the idea of kernels. Kernel functions have many nice combination properties. The sum or the product of existing valid kernels is again a valid kernel. With these combination properties, we are able to combine different kernels representing information from various sources. Zhao and Grishman [2005] use this property to combine different levels of text processing. They use kernels for tokenized -, sentence parsing - and deep dependency sources and combine them in order to overcome processing errors at a certain level of text processing. Inspired by this work, we want to combine kernels from different kind of features, namely our protein - and interaction features. We assume that when adding additional information for the proteins interacting with each other, we can improve performance, especially when the sentence talking about an interaction does not contain any interaction keywords or obvious features that express an interaction. In addition, consider sentence 4 from figure 5.6. Consider now the protein pair with the proteins BMP-2 and BMP-2. This instance has quite similar interaction features as the instances 1,2 or 3 with the important difference that this instance does not describe an interaction. In this case, it could happen that interaction features might be misleading and adding the information of the proteins could overcome the limitation of interaction features. As a consequence, we define one protein kernel K_P and one interaction kernel K_I. We combine these kernels through addition and create a combined kernel $K_{comb} = K_P + K_I$. When computing K_P, an instance of an Relation R is described by two protein feature vectors $\vec{p_1}$ and $\vec{p_2}$. K_P can be computed via:

$$K_{P[R_i,R_j]} = < \vec{p_{i1}}, \vec{p_{j1}} > \cdot < \vec{p_{i2}}, \vec{p_{j2}} > . \tag{5.1}$$

When considering interaction features, a relation R is represented by one interaction feature vector $\vec{r}$. Computing K_I is straightforward:

$$K_{I[R_i,R_j]} = < \vec{r_i}, \vec{r_j} > . \tag{5.2}$$

5.2.2 Evaluation

In total, we extract 5309 training instances, 1101 positive examples describing interaction between proteins and 4208 negative examples. As already mentioned, we skip parameter selection and validate our result using 10-fold cross-validation[5]. As done by Bunescu and Mooney [2005] we use the correct protein entities provided by the manual annotation. In table 5.15 we compare our various kernels with the subsequence kernel of Bunescu and Mooney [2005] and point to possible reasons for the performance difference. Since the results of the interaction kernel are already quite good, we present the effectiveness of different interaction features in table 5.16. The maximum number of true positives in our setting is 1101, which equals the number of true interactions in the data set. If the system correctly predicts that there is no interaction between two proteins, it is counted as true negative.

The extraction performance of our protein kernel K_P is in general not that satisfying. Note that protein entities pose a kind of challenge when extracting them from text in the

[5]Bunescu and Mooney [2005] also used 10-fold cross-validation, but with a different splitting

	K_P	K_I	K_{comb}	ERK
Precision	0.50	0.72	0.71	0.65
Recall	0.37	0.70	0.72	0.46
F-measure	0.43	0.71	0.72	0.54

Table 5.15: Comparison of the protein kernel K_P, interaction kernel K_I, the combined kernel K_{comb} and the subsequence kernel ERK from Bunescu and Mooney [2005]. Results were verified using 10-fold cross-validation.

training phase. If a protein like 'MMP12' is extracted from a sentence, a new protein feature vector is created, if 'MMP12' hasn't occurred yet in the training data. Remember from chapter 2 that proteins often suffer from alternative spellings (e. g. 'MMP-12', 'MMP12' or 'Mmp 12') and no common spelling convention does exist. Thus, we have to recognize these different spellings properly. Otherwise, for each alternative spelling, a new protein feature vector would be extracted. But these feature vectors would represent the same entity and would be similar, except for small variations in the protein name tokens. Clearly, this would introduce a lot of noise in the protein kernel. However, we cope with this problem. But how should we handle synonym names? 'EPOR' and 'erythropoietin receptor' also reflect the same protein entity, but this time the protein name token features will vary significantly. In addition, without an accurate synonym mapping file we are not able to disambiguate these two protein entities. One reason for the lower performance of the protein kernel is that our system currently does not handle the synonym case, thus introducing noise to our protein kernel (see chapter 6.2.2 for possible solutions for this problem). In addition, the protein pairs in this data set are very diverse. Thus, it is very difficult to extract significant patterns for proteins when interacting. Perhaps the inclusion of more training data could also help to overcome this problem. Furthermore, external knowledge like the HPRD molecular class or the GO code cannot be extracted for all proteins. There are 4141 protein mentions in the text data, resulting in 1017 proteins in our setting (no synonyms resolved). Only 60% of these proteins have a GO annotation. In total we have 146 different GO codes. 78 in the concept molecular function, 36 in cellular component and 32 in biological process. 50% of the 1017 proteins have a HPRD molecular class annotation. 73 different HPRD molecular classes can be found in the data set. Clearly,the quality of external features is heavily dependent on the number of proteins annotated already in external knowledge bases such as GO or HPRD.

The interaction kernel K_I performs quite well. Here we present only the results for unstemmed tokens, since stemming did not change the performance and also the number of support vectors was approx. equal. However, when including stemming, we have about 1000 token features less. We investigate the contributions of various features to the performance (see table 5.16). When using token features alone, the results are already very competitive. Since this is the only feature-based approach tried on the AImed corpus so far, we cannot compare the importance of token features from other systems. However, in the system from Zhou et al. [2005] in the ACE task, token features play a similar crucial

Tokens	•	•	•	•	•
Keywords		•	•	•	•
Overlap			•	•	•
Token Number				•	•
GO level					•
Accuracy	0.88	0.88	0.88	0.88	0.89
Precision	0.71	0.72	0.72	0.72	0.72
Recall	0.67	0.68	0.68	0.68	0.70
AUC	0.90	0.90	0.91	0.90	0.91
F-measure	0.69	0.70	0.70	0.70	0.71

Table 5.16: Performance of different features for the interaction kernel K_i

role, but their system makes use of a much larger set of features. The adding of keyword features yields a 1% improvement of the F-measure, indicating that our keyword list is not very representative for the AImed corpus. The overlap - and token between features, which were motivated in Ding et al. [2002] show no performance improvement. When adding the GO level feature, a recall improvement of 2% can be observed. Again, we are not able to extract for all interaction instances this external feature. The GO level feature can only be extracted for 48% of the instances. The distribution of the GO level feature over the positive and negative instances is approx. equal. A larger improvement of this feature can be expected with the ongoing progress of the annotation of gene products.

Our feature-based approach significantly outperforms the subsequence kernel from Bunescu and Mooney [2005]. This result coincides with the results from Zhou et al. [2005] in comparison with Bunescu and Mooney [2005] on the ACE RDC task. Bunescu and Mooney [2005] learn three different kernels for patterns which are representative for asserting a relationship between two entities. These three pattern kernels are then combined to one relation kernel. They make use of the following patterns:

- **Fore-Between:** Words before and between the two entities are responsible for expressing a relationship: 'interaction of P1 with P2'
- **Between:** Only the words between are responsible for the assertion of a relationship: 'P1 inhibits P2'
- **Between-After:** Words between and after are responsible for a relation: 'P1-P2 complex' or 'P1 and P2 interact'

Additionally, they heuristically restrict length and word positions of the patterns. For each pattern kernel, they exploit sparse subsequences of a rich set of linguistic features (i. e. words, POS tags, chunks or WordNet synsets). First of all, the restriction to a certain length in the Between-pattern may not be very reasonable (see e. g. sentence 5 in figure 5.6). This restriction seems to be more intuitive in the Fore-Between pattern and in the

Between-After pattern (we also use a similar kind of restriction in that case). It is also not clear, how self-interactions (see sentence 3 from figure 5.6) can be extracted with the patterns described above. However, Bunescu and Mooney [2005] completely lexicalize the patterns mentioned above and thus these patterns are very sparse, when considering only words. To alleviate this problem they introduce the linguistic features mentioned above and combine them, resulting in an incredible large feature space. Recall that in our setting the problem of relation detection is casted here to text classification. Approaches to text categorization based on rich linguistic features have obtained less accuracy than the traditional bag-of-words approaches (e. g. Cornelis et al. [2003]).

5.2.3 Summary

We have shown that feature-based relation extraction shows very competitive results on extracting protein-protein interactions from text. Again, we did not make use of any sophisticated NLP knowledge. We introduced the idea of using external knowledge base features (especially ontology features). These ontology features seem to be helpful and we expect more effectiveness on these features with ongoing annotation of gene products with GO. We used different views of describing an interaction instance and thus resulted in different kernels, which were combined in order to overcome processing errors of single kernels. However, combining these different kernels could not improve performance significantly, due to (1) the high performance of interaction features and (2) the data sparseness problem for protein entities. In this experiment, we assume that protein names are already given. In the future we plan to incorporate named entity recognition to this setting. First, the protein names would be identified by a CRF and second, we would apply our trained SVM to classify relations. As a matter of course, a performance decrease is expected due to mislabeled protein entities. Nonetheless, the performance should not be affected so heavily, because of boundary errors of the protein tokens. A larger difficulty is the handling of protein families. Our CRF system is currently trained to recognize also these families, but the annotators of the AImed corpus discard protein families. This could results in a higher number of false positives. Section 6.2.2 will outline future work on this topic.

Chapter 6

Conclusion

6.1 Summary

The goal of this thesis was to address the problem of named entity recognition and the problem of relation extraction from biomedical text corpora. Hereby, we concentrated on supervised machine learning.

As already summarized in section 5.1.5, CRFs are a suitable probabilistic model for named entity recognition for various domains and different languages. We show this by evaluating CRFs across different domains and different languages. We are able to tag a rich set of biomedical entities with high performance, even when different entities often have a similar text representation (e. g. DNA and protein entities). Applying a conditional random field model to unseen text phrases turns out to be very effective, once the expensive training of the model is done. We highlight this property by the implementation of a text analysis pipeline with a graphical user interface on top of that pipeline (see figures 1.1 and 1.3).

A very challenging task in this thesis was the relation detection task for proteins (see section 5.2). This research topic is still very new and especially in the biomedical domain only little work does exist. We focused on relation detection and treated this problem as classification problem. Hereby, we made use of different (also external) knowledge bases. The use of ontology features was introduced by us and seems to be useful. However, combining diverse kernels generated from various sources (referred to as protein - and interaction kernels) could not help to improve performance significantly. This is mainly due to the data sparseness problems of the protein entities (see section 6.2.2 for future work). Our bag-of-word related approach shows very competitive results on the AImed corpus.

6.2 Outlook

6.2.1 Named entity recognition

We will start with ideas on improving named entity recognition. The question is: What will it take to improve performance on named entity recognition? As could be seen from alternative scoring schemes (softer matching criteria), a lot of mistakes are due to wrong boundary recognition. The common way to reduce this type of error is to include a post-processing step to fine-tune boundaries. However, it would be much more desirable, if boundary errors would occur much less a priori. Remember from section 2.2 that we treated the problem of named entity recognition as a task of segmenting and labeling sequential data. Therefore, we only applied a single tagging scheme in this thesis. Tokens could be either outside (**O**) or at the beginning (**B-t**) or inside (**I-t**) an entity. **t** indicates thereby the type of entity. In the future we want to conduct experiments for named entity recognition, where we make use of different tagging schemes. What happens when we introduce an additional state, indicating the end of an entity (**E-t**)? Especially for biomedical named entity recognition, where the average entity length often exceeds one (remember the histogram from figure 5.1), this could have a severe impact on making less right boundary errors. Another obvious scheme is just to have one label which marks whether we are inside an entity or not. This idea can be extended to an extensive analysis for different tagging schemes. Incorporating different tagging schemes could also result in a further optimization step for creating a model. Another way to improve the performance on named entity recognition is with the help of new features. Zhou et al. [2005] conclude that in their relation detection task deeper syntactic knowledge could not boost performance, but text chunking features did. In most of the existing NER systems POS-tags are usually exploited but text chunks are not. Perhaps the use of text chunks could also improve NER performance. Another question is: Why are external resources not more useful? As we already figured out, for some domains there are no complete dictionaries, therefore their usefulness is naturally limited. But is there another way to incorporate dictionary features in the CRF approach? One could try to set up a kind of similarity measure between candidate entities and dictionary members. If the similarity measure exceeds a certain threshold, this feature is put on. In addition, we want to use the approach of CRFs to tag more type of entities in the future. Our CRF model has already been extended to tag diseases and the preliminary results are very satisfactory. Further entities could be chemical compounds or drugs, e. g. . Biomedical named entity recognition remains a challenging task and the BioCreAtIvE II workshop[1] which will be held in April 2007 will hopefully give new insights in this exciting research topic.

6.2.2 Relation Extraction

As highlighted in chapter 2 gene and protein names pose a lot of challenges when including them in a relation extraction system. Besides different spelling alternatives, there are

[1] http://biocreative.sourceforge.net/index.html

in average five synonyms for one given gene. Indeed, we handled the different spelling alternatives in this theses, but we did not handle the synonym problem. In the future, we also want to cope this problem. As a consequence, the data sparseness will be reduced significantly and we expect improvement, when adding the protein kernel into our system. Furthermore, we want to try our approach on several other data sets to get a deeper insight of the representation of proteins in different data sets. We also think about a higher-order representation for proteins, which would additionally reduce data sparseness. However, the problem still remains, that a lot of proteins are not annotated yet and thus it will be difficult to find such a higher order for every protein. In the future, we also want to exploit shallow linguistic features such as text chunks, which have been found to be very useful in relation extraction [Zhou et al., 2005]. However, the need for a standard benchmark set for protein-protein interaction extraction is apparent, but the BioCreAtIvE II evaluation in April 2007 will provide such a protein interaction extraction task. Hopefully, this data set will develop to the strongly needed standard benchmark set.

A very new and promising direction was recently introduced by Culotta et al. [2006]. They treat the relation extraction task as sequential labeling task (see section 2.2) and integrate pattern discovery in the relation extraction task. Thus, they can apply a CRF to this task and report a performance improvement, due to the integration of contextual and relational patterns. However, at the moment they are not able to enumerate all pairs of entities within a sentence, because their approach needs predominantly positive examples. As a consequence, they can apply their approach only to biographical texts, where there is always one principal entity. This does not hold for protein interactions, but nevertheless this new direction is a first step to a very efficient solution of the relation extraction problem. In addition, when mining other kinds of relations in the biomedical area, where a principal entity does exist (e. g. disease-gene relations), this model can be a very suitable and elegant choice.

Appendix A

Data and Software

The provided compact disc provides data from the evaluations. Software used and implemented in this theses can also be found on the disc. Part of the software, developed for this thesis, is an XML-based framework which allows to efficiently define and train conditional random fields, based on MALLET[1]. In addition, a text processing pipeline based on LingPipe[2], which includes sentence splitting and tokenization, has been implemented. A simple graphical user interface has been developed to show the efficiency of named entity recognition with conditional random fields. These methods and demos are all implemented in Java. Further information is provided within the documentation of the software.

[1] http://mallet.cs.umass.edu
[2] http://www.alias-i.com/lingpipe/

Bibliography

R. Bunescu, R. Ge, R. J. Kate, E. M. Marcotte, R. J. Mooney, A. K. Ramani, and Y. W. Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, February 2005. doi: 10.1016/j.artmed.2004.07.016.

R. C. Bunescu and R. J. Mooney. Subsequence kernels for relation extraction. In *Proceedings of the 19th Conference on Neural Information Processing Systems*, 2005.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

K. B. Cohen and L. Hunter. *Natural language processing and systems biology*, pages 147–174. Springer, December 2004.

H. Cornelis, A. Koster, and M. Seutter. Taming wild phrases. In *Advances in Information Retrieval, 25th European Conference on IR Research (ECIR2003)*, pages 161–176, 2003.

M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, 1999.

A. Culotta, A. Mccallum, and J. Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Human Language Technology Conference/North American chapter of the Association for Computational Linguistics Annual Meeting*, June 2006.

J. N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. In *The Annals of Mathematical Statistics*, pages 1470–1480, 1972.

J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. Mining medline: abstracts, sentences, or phrases? In *Pac Symp Biocomput*, pages 326–337, Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, USA., 2002.

F. Erik, T. K. Sang, and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In W. Daelemans and M. Osborne, editors, *Proceedings of the Conference on Natural Language Learning (CoNLL-2003)*,

pages 142–147, Edmonton, Kanada, 2003. Association for Computational Linguistics (ACL).

J. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair, and C. Manning. Exploiting context for biomedical entity recognition: From syntax to the web. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, 2004.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*, 2003.

K. Fukuda, A. Tamura, T. Tsunoda, and T. Takagi. Toward information extraction: identifying protein names from biological papers. *Pac Symp Biocomput*, pages 707–718, 1998.

R. Grishman and B. Sundheim. Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message Understanding*, November 1995.

J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. 1971.

D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear svm-based feature selection. In *ICML '04: Twenty-first international conference on Machine learning*. ACM Press, 2004. ISBN 1581138285. doi: 10.1145/1015330.1015421.

M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, , P. Tonellato, P. Jaiswal, T. Seigfried, and R. White. The gene ontology (go) database and informatics resource. *Nucleic Acids Res*, 32(Database issue), January 2004. ISSN 1362-4962.

L. Hirschmann. Various criteria in the evaluation of biomedical named entity recognition.

E. T. Jaynes. Information theory and statistical mechanics - jaynes. *The Physical Review*, 106(4):620–630.

T. Karopka, S. Scheel, S. Bansemer, and A. Glass. Automatic construction of gene relation networks using text mining and gene expression data. *Medical Informatics and the Internet in Medicine*, 29:169–183, June 2004.

J. D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. Genia corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1, 2003. ISSN 1367-4803.

J. D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. Introduction to the bioentity recognition task at jnlpba. In *Proceedings of Joint Workshop on Natural Language Processing in Biomedicine and its applications*, 2004.

M. Krauthammer, A. Rzhetsky, P. Morozov, and C. Friedman. Using blast for identifying gene and protein names in journal articles. *Gene*, 259(1-2):245–252, December 2000. ISSN 0378-1119.

J. Lafferty, A. Mccallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

L. Lin, T. Tsai, W. Chou, K. Wu, T. Sung, and W. Hsu. A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th workshop on data mining in bioinformatics*, pages 56–61, 2004.

R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*, pages 49–55, 2002.

A. Mccallum and W. Li. Early results for named entity recognition with conditional random fields. In *Proceedings of CoNLL-2003*, 2003.

R. Mcdonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, 6 Suppl 1, 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-S1-S6.

C. Nedellec. Learning language in logic - genic interaction extraction challenge. In *Proceedings of the ICML-2005 Workshop on Learning Language in Logic (LLL05)*, pages 31–37, 2005.

T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, February 2001. ISSN 1367-4803.

S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. K. Gandhi, M. Gronborg, N. Ibarrola, N. Deshpande, K. Shanker, H. N. Shivashankar, B. P. Rashmi, M. A. Ramya, Z. Zhao, K. N. Chandrika, N. Padma, H. C. Harsha, A. J. Yatish, M. P. Kavitha, M. Menezes, D. R. Choudhury, S. Suresh, N. Ghosh, R. Saravana, S. Chandran, S. Krishna, M. Joy, S. K. Anand, V. Madavan, A. Joseph, G. W. Wong, W. P. Schiemann, S. N. Constantinescu, L. Huang, R. Khosravi-Far, H. Steen, M. Tewari, S. Ghaffari, G. C. Blobe, C. V. Dang, J. G. Garcia, J. Pevsner, O. N. Jensen, P. Roepstorff, K. S. Deshpande, A. M. Chinnaiyan, A. Hamosh, A. Chakravarti, and A. Pandey. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res*, 13(10):2363–2371, October 2003. ISSN 1088-9051. doi: 10.1101/gr.1680803.

A. K. Ramani, R. C. Bunescu, R. J. Mooney, and E. M. Marcotte. Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biol*, 6(5), 2005. ISSN 1465-6914. doi: 10.1186/gb-2005-6-5-r40.

S. Ray and M. Craven. Representing sentence structure in hidden markov models for information extraction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, 2001.

T. C. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proceedings of Pacific Symposium on Biocomputing*, pages 517–528, 2000.

B. Rosario and A. Hearst. Multi-way relation classification: Application to protein-protein interaction. In *Human Language Technology Conference on Empirical Methods in Natural Language Processing*, 2005.

B. Rosario and M. Hearst. Classifying semantic relations in bioscience texts. In *Proceedings of the Annual Meeting of Association of Computational Linguistics (ACL '04)*, 2004.

G. Salton. *Automatic text processing: the transformation, analysis and retrieval of information by computer*. 1989.

B. Settles. Biomedical named entity recognition using conditional random fields. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, 2004.

L. Shi and F. Campagne. Building a protein name dictionary from full text: a machine learning term extraction approach. *BMC Bioinformatics*, 6(1), April 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-88.

L. Tanabe, N. Xie, L. H. Thom, W. Matten, and W. J. Wilbur. Genetag: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6 Suppl 1, 2005. ISSN 1471-2105. doi: 10.1186/1471-2105-6-S1-S3.

J. M. Temkin and M. R. Gilder. Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19(16):2046–2053, November 2003. ISSN 1367-4803.

T.-H. H. Tsai, S.-H. H. Wu, W.-C. C. Chou, Y.-C. C. Lin, D. He, J. Hsiang, T.-Y. Y. Sung, and W.-L. L. Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics*, 7(1), February 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-92.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. ISBN 0471030031.

C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, May 2002. ISSN 0028-0836. doi: 10.1038/nature750.

H. M. Wallach. Conditional random fields: An introduction. Technical report, 2004.

J. A. White, P. J. Mcalpine, S. Antonarakis, H. Cann, J. T. Eppig, K. Frazer, J. Frezal, D. Lancet, J. Nahmias, P. Pearson, J. Peters, A. Scott, H. Scott, N. Spurr, C. Talbot, and S. Povey. Guidelines for human gene nomenclature (1997). hugo nomenclature committee. *Genomics*, 45(2):468–471, October 1997. ISSN 0888-7543.

I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg. Dip: the database of interacting proteins. *Nucleic Acids Res*, 28(1):289–291, January 2000. ISSN 0305-1048.

Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. In *Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence (UAI 2006)*, 2006.

A. Yeh, A. Morgan, M. Colosimo, and L. Hirschman. Biocreative task 1a: gene mention finding evaluation. *BMC Bioinformatics*, 6, 2005.

X. Zhang, X. Lu, Q. Shi, X. Q. Xu, H. C. Leung, L. N. Harris, J. D. Iglehart, A. Miron, J. S. Liu, and W. H. Wong. Recursive svm feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7, 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-197.

S. Zhao. Name entity recognition in biomedical text using a hmm model. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA))*, 2004.

S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 419–426, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the ACL*, 2005.

G. D. Zhou and J. Su. Exploring deep knowledge resources in biomedical name entity recognition. In *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, 2004.