April 2014

Investigating Genotype-Phenotype relationship extraction from biomedical text

Maryam Khordad
The University of Western Ontario

Supervisor
Robert E. mercer
The University of Western Ontario

Graduate Program in Computer Science

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

INVESTIGATING GENOTYPE-PHENOTYPE RELATIONSHIP EXTRACTION FROM BIOMEDICAL TEXT

(Thesis format: Monograph)

by

Maryam Khordad

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

During the last decade biomedicine has developed at a tremendous pace. Every day a lot of biomedical papers are published and a large amount of new information is produced. To help enable automated and human interaction in the multitude of applications of this biomedical data, the need for Natural Language Processing systems to process the vast amount of new information is increasing. Our main purpose in this research project is to extract the relationships between genotypes and phenotypes mentioned in the biomedical publications. Such a system provides important and up-to-date data for database construction and updating, and even text summarization.

To achieve this goal we had to solve three main problems: finding genotype names, finding phenotype names, and finally extracting phenotype–genotype interactions. We consider all these required modules in a comprehensive system and propose a promising solution for each of them taking into account available tools and resources.

BANNER, an open source biomedical named entity recognition system, which has achieved good results in detecting genotypes, has been used for the genotype name recognition task.

We were the first group to start working on phenotype name recognition. We have developed two different systems (rule-based and machine-learning based) for extracting phenotype names from text. These systems incorporated the available knowledge from the Unified Medical Language System metathesaurus and the Human Phenotype Onotolgy (HPO). As there was no available annotated corpus for phenotype names, we created a valuable corpus with annotated phenotype names using information available in HPO and a self-training method which can be used for future research.

To solve the final problem of this project i.e. , phenotype–genotype relationship extraction, a machine learning method has been proposed. As there was no corpus available for this task and it was not possible for us to annotate a sufficiently large corpus manually, a semi-automatic approach has been used to annotate a small corpus and a self-training method has been proposed to annotate more sentences and enlarge this corpus. A test set was manually annotated by an expert. In addition to having phenotype-genotype relationships annotated, the test set con-

tains important comments about the nature of these relationships. The evaluation results related to each system demonstrate the significantly good performance of all the proposed methods.

Keywords: Relation Extraction, Named Entity Recognition, Phenotype, Genotype, Semi-supervised learning

Acknowlegements

I would like to express my special appreciation and thanks to my supervisor Professor Mercer you have been a tremendous mentor for me. Your support and advice on my research was priceless. It was an honor for me to work under your supervision and I really believe I was lucky for that.

I would like to express the deepest appreciation to Professor Rogan, who suggested my thesis topic. Without his guidance and persistent help this project would not have been possible.

I would also like to thank my labmates Shifta Ansari and Rushdi Shams for all their help and support. In addition, a thank you to Syeed Ibn Faiz who provided me with his tools.

I would like to thank the University of Western Ontario, the Faculty of Science and the Computer Science Department for providing me financial support as a Teaching Assistant and a Research Assistant.

A special thanks to my family. Words cannot express how grateful I am to my mother, father, and brothers for all of the sacrifices that you have made on my behalf. At the end I would like express appreciation to my beloved husband Reza who was always my support in the moments when there was no one to answer my queries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the last decade biomedicine has developed at a tremendous pace. Every day a tremendous number of biomedical papers are published and a large amount of new information is produced. To help enable automated and human interaction in the multitude of applications of this biomedical data, the need for Natural Language Processing (NLP) systems to process the vast amount of new information is increasing. Current NLP systems try to extract from the biomedical literature different knowledge such as: protein–protein interactions [13, 38, 45, 63, 76, 117], new hypotheses [47, 48, 105], relations between drugs, genes, and cells [36, 89, 106], relations between genes and diseases [25, 88], protein structure [39, 51], and protein function [6, 109].

One of the important pieces of information contained in biomedical literature is the newly discovered relationships between phenotypes and genotypes. The genotype refers to the entire set of genes in a cell, an organism, or an individual. Phenotypes are the observable characteristics of a cell or organism, including the result of any test that is not a direct test of the genotype [103]. A phenotype of an organism is determined by the interaction of its genetic constitution and the environment. Skin colour, height, and behaviour are some examples of phenotypes. It is worth noting that throughout this thesis we do not take into account the phenotypes at the cellular level (inside the cell).

A lot of research experiments are being performed to discover the role of DNA sequence variants in human health and disease. The results of these experiments are published in the

biomedical literature. Experts want to know if a disease is caused by a genotype or a special genotype determines certain characteristics in people. This information is very valuable for biologists, physicians, and patients. There are some resources which collect this information and visualize it for people who are interested. Because of the large quantity of information, a reliable automatic system to extract this information for future organization is desirable. Such a system provides important and up-to-date data for database construction and updating, and even text summarization.

Our purpose in this research project is to make an automatic system which extracts the relationships between phenotypes and genotypes from biomedical papers. Assume that we have the following sentence in a research paper:

- Mutations of the MECP2 gene at Xq28 are associated with Rett syndrome in females and with syndromic and nonsyndromic forms of mental retardation in males.

Our final system should be able to extract the following information from this sentence.

1. *Mutations of the MECP2 gene at Xq28* is a genotype.
2. *Mental retardation* and *Rett syndrome* are phenotypes.
3. According to this sentence there are two relationships in this sentence between the mentioned genotype and two phenotypes.

Therefore our system must have the ability to accomplish three different tasks: recognizing phenotype names, recognizing genotype names and finally detecting the relationships between them according to the sentences. Throughout this thesis we will explain our proposed methods for each of these fundamental tasks.

Phenotype name recognition is a very difficult and complicated task. According to our knowledge we are the first group to start working on this topic. We proposed a rule-based system [55] and a machine learning-based system [56] to solve this problem and a method for improving the results in the machine-learning based system. So far, we are aware of only one other published paper [21] on this subject.

Recognizing gene and protein names in biomedical literature is a well-studied problem. In the previous years the BioCreAtIvE challenge evaluation[1] [1] had several competitions in this area. Many groups participated in these competitions and some very good systems were proposed. We decided to use one of the available systems named BANNER [62] in our own system to extract genotype names from biomedical text.

Extracting relationships between phenotypes and genotypes is the last step in this project. We are not aware of any other work on this topic. We proposed a machine learning based approach to attack this problem. As there was no available corpus for training our machine learning model we generated a corpus semi-automatically and using a semi-supervised learning method we improved our final results.

The text from which the relations are extracted has been drawn from the full-text articles from PubMed (2009) and BioMedCentral (2004), the abstracts obtained from querying PubMed [3], and abstracts from Phenominer [21].

The remainder of the thesis is organized as follows: Chapter 2 describes a general relation extraction system, its modules and previous works related to each module. Chapter 3 explains the idea behind semi-supervised learning and its algorithms. Chapter 4 gives a brief introduction to Conditional Random Fields (CRF), a machine learning algorithm for labelling sequence data. Chapter 5 gives a short description of NLP tools that have been used in this project. Chapter 6 is devoted to BANNER, the tool we applied to the genotype recognition task. Chapter 7 describes a rule-based method we proposed for phenotype name recognition. Chapters 8 and 9 explain a machine learning-based method for phenotype name recognition and a suggestion for improving it. Our phenotype-genotype relation extraction system is described in Chapter 10 . Finally we conclude the work presented in this thesis in Chapter 11 and discuss directions for possible future work.

[1] BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology) challenge is an effort for evaluating text mining and information extraction systems applied to the biomedical domain.

Chapter 2

A General Relation Extraction System

The ultimate goal of this thesis is to extract phenotype-genotype relations from biomedical text. Many systems have been developed to find the relationships between different entities from biomedical text. Some of these systems extract the relationships between homogeneous entities like protein-protein interactions [13, 38, 45, 63, 76, 117] and some others suggest methods to extract relations between heterogeneous entities like relations between drugs, genes, and cells [36, 89, 106], or relations between genes and diseases [25, 88]. Figure 2.1 [118] illustrates a general block diagram for a system which finds the relationship between biomedical entities. This system is composed of different modules. The remainder of this chapter provides a general discussion of each of these modules with some details related to the focus of this thesis.

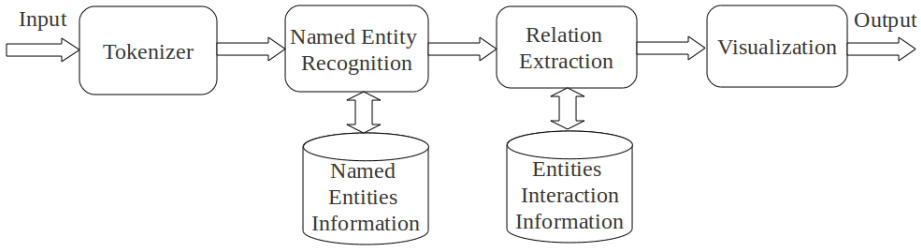

Figure 2.1: General block diagram for finding relationships between entities

2.1 Tokenizer

The first module is the tokenizer. Tokenization is the process of breaking the text up into its constituent units or its tokens. Tokens may vary in granularity depending on the particular application. Consequently, chapters, sections, paragraphs, sentences, words, syllables, or phonemes can be used as tokens. Many different algorithms exist for breaking up the text into any level of tokenization.

Ding et al. [31] applied different levels of tokenization such as phrases, sentences, and abstracts from MEDLINE to find their effectiveness in the application of mining interactions between biochemical entities based on co-occurrences. Experimental results showed that abstracts, sentences, and phrases can produce comparable extraction results, with sentences being more effective than phrases and being as effective as abstracts. In this work we have tokenized full-text articles and abstracts at the sentence and word level.

In breaking up a text into sentences, the most challenging part is distinguishing between a period that shows an end of sentence and a period that is a part of a previous token like the abbreviations ”Mr.”, ”Dr.”, etc. This is a well-studied problem. MedPost [100] is one of the excellent performing and freely available software solutions that deal with biomedical text. MedPost was originally designed for MEDLINE abstracts. A modified version was developed in Dr. Mercer's Cognitive Engineering Laboratory to work with full-text articles and was named PostMed to distinguish it from the original.

2.2 Named entity recognition

Named entities are phrases that contain the names of people, companies, cities, etc., and specifically in biomedical text entities such as genes, proteins, diseases, drugs, or organisms. Consider the following sentence as an example:

- The RPS19 gene is involved in Diamond-Blackfan anemia.

There are two named entities in this sentence: *RPS19 gene* and *Diamond-Blackfan anemia*.

Named Entity Recognition (NER) is the task of finding references to known entities in natural language text. An NER technique may consist of some natural language processing methods like part-of-speech (POS) tagging and parsing.

An NER system has to deal with three different problems: the recognition of a named entity in text, assigning the named entity to a predefined class (gene, protein, drug, etc), and finding the most suitable name for the entity if there are some synonyms for naming the entity [65]. The latter is especially important if the recognized entities are to be combined with information from other resources, such as databases and ontologies.

Over the past years it has turned out that finding the names of biomedical objects in literature is a difficult task. Some problematic factors are: the existence of millions of entity names, a constantly growing number of entity names, the lack of naming agreement, an extreme use of abbreviations, the use of numerous synonyms and homonyms, and the fact that some biological names are complex names that consist of many words, like "increased erythrocyte adenosine deaminase activity". Even biologists do not agree on the boundary of the names [65].

Named Entity Recognition in the biomedical domain has been extensively studied and, as a consequence, many methods have been proposed. Some methods like MetaMap [7] and mgrep [28] are generic methods and find all kinds of entities in the text. Some methods, however, are specialized to recognize particular types of entities like gene or protein names [39, 58], diseases and drugs [89, 94, 114], mutations [46] or properties of protein structures [39].

NER techniques are usually classified into three categories [65]. Dictionary-based techniques are used by, for example, Krauthammer et al. [58] to match phrases from the text against some existing dictionaries. Rule-based techniques used by, for example, Fukuda et al. [37] make use of some rules to find entity names in the text. Machine learning techniques as, for example, used by Nobata et al. [79] transform the NER task into a classification problem.

2.2.1 Dictionary-based techniques

Dictionaries are large collections of entity names. Exactly matching words and phrases in texts against entity names in a dictionary is a very precise NER method but it cannot find

all named entities and yields low recall [1] [65]. To improve this method some researchers use inexact matching techniques, or try to generate typical spelling variants for each name and add them to the available dictionary. Then they use this new dictionary to find exact matches against text words. Swiss–Prot [9] is a biological database of protein sequences which is widely used for protein NER. Flybase [40], the Unified Medical Language System (UMLS) [50], the Pharmacogenetics Knowledge Base (PharmGKB) [57], and the Online Mendelian Inheritance in Man (OMIM) [73] are examples of other biomedical databases. Coulet et al. [25] use PharmGKB [57] to extract genes, drugs, and phenotypes in the literature.

The Human Phenotype Ontology (HPO) [90] is an ontology that tries to provide a standardized vocabulary of phenotypic abnormalities encountered in human disease. The HPO was constructed using information initially obtained from the Online Mendelian Inheritance in Man (OMIM) [73] after which synonym terms were merged and the hierarchical structure was created between terms according to their semantics. The hierarchical structure in the HPO represents the subclass relationship. The HPO currently contains over 9500 terms describing phenotypic features.

The Human Genome Nomenclature (HUGO) [110] contains a list of approved human gene names and symbols. In addition this list contains some symbols and names that are used before approval. The HUGO Gene Nomenclature Committee assigns a unique symbol for each gene. It has approved over 29,000 human gene symbols and names. The committee tries to facilitate communication and electronic information retrieval of human genes avoiding any conflicts among gene symbols.

The Gene Ontology (GO) [8] contains defined terms representing gene product properties. This ontology covers three domains: cellular component, molecular function, and biological process. This ontology consists of many databases, including several of the world's major repositories for plant, animal, and microbial genomes.

The GENIA corpus [80] is a collection of 2000 Medline abstracts which were collected

[1] Recall is the percentage of correct entity names found compared to all correct entity names in the corpus and can be used as a measure of completeness.

A	AAAC	**E**	AACG	**I**	AAGT	**M**	ACAC	**Q**	ACCG	**U**	ACGT	**Y**	AGAG
B	AAAG	**F**	AACT	**J**	AATC	**N**	ACAG	**R**	ACCT	**V**	ACTC	**Z**	AGAT
C	AAAT	**G**	AAGC	**K**	AATG	**O**	ACAT	**S**	ACGC	**W**	ACTG		
D	AACC	**H**	AAGG	**L**	AATT	**P**	ACCC	**T**	ACGG	**X**	ACTT		

0	AGCC	**4**	AGGG	**8**	AGTT	**/**	ATCC	**,**	ATCC	**?**	ATCC	**-**	ATCC
1	AGCG	**5**	AGGT	**9**	ATAT	****	ATCC	**;**	ATCC	**"**	ATCC		
2	AGCT	**6**	AGTC	**]**	ATCC	**(**	ATCC	**:**	ATCC	**.**	ATCC		
3	AGGC	**7**	AGTG	**[**	ATCC	**)**	ATCC	**!**	ATCC	**space**	ATCC		

Figure 2.2: Conversion Table [59]

using the three MeSH [2] terms, "human", "blood cells", and "transcription factors". This corpus has been annotated for biological terms and can be used to train entity extraction algorithms.

Krauthammer et al. [59] use BLAST as an inexact matching algorithm. The BLAST [5] programs are capable of searching similarities between sequences in protein and DNA databases. In this approach, they translate a list of gene and protein names extracted from GenBank[3] [2] into an alphabet of DNA sequences using a conversion table (Figure 2.2). Each character of the name is substituted with a unique nucleotide string. As an example, the gene name zgap1 would be translated into: AGATAAGCAAACACCCAGCG. Then they translate the scientific papers into a continuous string of nucleotides using the same table. For example, the sentence "For instance ErbB, Ras and Raf all lie on the ERK MAP kinase (MAPK) pathway." would be represented by Figure 2.3. Finally they search the whole paper string to find similarities between nucleotide strings of gene and protein names in the standard BLAST library and translated papers. A BLAST output file contains the list of these similarities and their corresponding protein and gene names [59].

[2]Medical Subject Headings (MeSH) is a comprehensive controlled vocabulary for the purpose of indexing journal articles and books in the life sciences; which is used by the MEDLINE/PubMed article database.

[3]The GenBank sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations.

```
AACTACATACCTATCCAAGTACAGACGCACGGAAACACAGAAATAACGATCCATCCAAC
GACCTAAAGAAAGATCCATCCACCTAAACACGCATCCATCCAAACACAGAACCATCCAC
CTAAACAACTATCCAAACAATTAATTATCCAATTAAGTAACGATCCACATACAGATCCA
CGGAAGGAACGATCCAACGACCTAATGATCCACACAAACACCCATCCAATGAAGTACAG
AAACACGCAACGATCCATCCACACAAACACCCAATGATCCATCCATCCACCCAAACACG
GAAGGACTGAAACAGAG
```

Figure 2.3: Example

2.2.2 Rule-based techniques

Rule-based approaches (e.g. , [37, 55, 88]) make use of lexical and linguistic rules to find entity names in the text. In early systems, rules were created manually using human experts. These rules specify the characteristics of named entities and their context, for example, surface clues (capital letters, symbols, digits). Using these rules, some candidate entity names are chosen and then these candidates can be expanded using some syntactic rules. For example, some functional terms surrounding a candidate are included in the protein name (such as *kinase* or *receptor*). A part-of-speech (POS) tagger may be used to provide grammatical data for further expansion rules. For instance, if a part of a nominal phrase is identified as a named entity the whole nominal phrase can be recognized as a named entity.

One problem in rule-based systems is that they are not robust enough to recognize previously unseen names. In addition, the process of writing rules manually is too time-consuming and labourious. The precision[4] and recall[5] in these systems depend on how specific the rules are. Usually the rules cover special situations and systems achieve high precision and low recall.

2.2.3 Machine learning techniques

In this approach the NER task is transformed to a classification problem. Dictionary matches, word occurrence, context, and word morphology are used as features. Here, Support Vec-

[4]Precision is the percentage of correct entity names as determined by a human in all entity names found and can be seen as a measure of soundness.

[5]Recall is the percentage of correct entity names as determined by a human found compared to all correct entity names in the corpus and can be used as a measure of completeness.

tor Machines (SVM) [75], Hidden Markov Models [86], Conditional Random Fields [61] and naïve Bayes [75] are broadly and successfully applied. ABGene [107], BANNER [62], AB-NER [96] and LingPipe [15] are some examples of machine learning-based NER methods in the biomedical domain.

2.3 Relation extraction

The extraction of relations between biomedical objects has attracted much attention and several different approaches have been proposed. Generally, current approaches can be divided into three categories [118]: Computational linguistics-based, rule-based, and machine learning and statistical methods. Furthermore some systems ([13, 38, 88], for instance) have combined these approaches and have proposed hybrid methods.

2.3.1 Computational linguistics-based methods

These methods define grammars to describe the syntax of sentences and make use of parsers to find the grammatical structure of sentences and dependency relations among words. In the first step of these methods the corpus is parsed to find the structure of each sentence and find the relationships between different phrases in a sentence. Using this information is very useful to extract relations between protein and gene names. However, efficient and accurate parsing of unrestricted text (like biomedical texts) is beyond the capability of current techniques. It is not applicable to use standard parsing algorithms because they are too expensive to use on very large corpora and they are not robust enough [98]. For example the BLLIP parser [17] spends about 1 minute to parse a sentence and its F-score [6] is reported to be 91.02 [17] on section 23 of the Penn Treebank [68]. An alternative way is to use shallow parsers [53]. In shallow parsing techniques, the sentence is not parsed completely. Instead, the sentence is broken up into non-overlapping word sequences or phrases, such that syntactically related words are grouped together.

[6] The harmonic mean of equally weighted precision and recall.

Shallow parsing approaches

Shallow parsers decompose each sentence partially into some phrases and find the local dependencies between phrases. Each phrase is tagged using a set of predefined grammatical tags such as: Noun Phrase, Verb Phrase, Prepositional Phrase, Adverb Phrase, Subordinate Clause, Adjective Phrase, Conjunction Phrase, and List Marker.

Leroy et al. [64] develop a shallow parser to extract relations between entities from abstracts. The type of these entities has not been restricted. They start from a syntactic perspective and extract relations between all noun phrases regardless of their type. Their parser looks for certain patterns in the text based on English closed-class words, e.g. , conjunctions and prepositions. Closed-class words do not change over time. By using them, the resulting templates are general and do not depend on a pre-specified biomedical vocabulary. The parser is composed of four cascaded finite state automata (FSA) to capture the content of paper abstracts and then find the structure of the content: the FSA for Basic Sentences, the FSA for the preposition *of*, the FSA for the preposition *by* and the FSA for the preposition *in*. The recognized patterns in each FSA contain the binary relations between two noun phrases. Each relation can contain up to five elements and requires a minimum of two elements.

Interacting proteins usually have similar biological functions. He and DiMarco [45] base their method on this characteristic of protein-protein relationships. In this paper the idea of lexical chaining has been used. While looking for a protein-protein interaction they focus on finding some biological terms about the common functions of two interacting proteins in the context surrounding the protein-protein interaction. If such terms are found, there is strong evidence that the interaction is a valid biological interaction.

SemGen [88] identifies and extracts causal interaction of genes and diseases from MEDLINE citations. Texts are parsed using MetaMap. The semantic type of each noun phrase tagged by MetaMap is the basis of this method. Twenty verbs (and their nominalizations) plus two prepositions, *in* and *for*, are recognized as indicators of a relation between a genetic phenomenon and a disorder.

Sekimizu et al. [95] use a shallow parser to find noun phrases in the text. The most fre-

quently seen verbs in the collection of abstracts are believed to express the relations between genes and gene products. Based on these noun phrases and frequently seen verbs, the subject and object of the interaction are recognized.

Obviously, shallow parsers extract simple binary relationships successfully but when there are more complex relations among more than two entities they usually have erroneous results. Deep parsers yield better and more precise results but they are expensive to use in terms of time.

Deep parsing approaches

Systems based on deep parsing determine the complete syntactic structure of a sentence or a string of symbols in a language and therefore are potentially more accurate. Deep parsing approaches are divided into two main types according to the way of constructing grammars: rationalist methods and empiricist methods. Rationalist methods define grammars manually, while empiricist methods generate the grammar automatically using some observations.

Coulet et al. [25] propose a method to capture pharmacogenomics (PGx) relationships and build a semantic network based on relations. They use lexicons of PGx key entities (drugs, genes, and phenotypes) from PharmGKB[57] to find sentences mentioning pairs of key entities. Using the Stanford parser [30] these sentences are parsed and their dependency graphs[7] are produced. According to the dependency graphs and two patterns, the subject, object, and the relationship between them are extracted.

RelEx[38] makes dependency parse trees from the text and applies a small number of simple rules to these trees to extract protein–protein interactions. We will explain RelEx later in Section 2.3.2 because it is a rule-based method too. Temkin and Gilder [108] use a lexical analyzer and a context free grammar to make an efficient parser to capture interactions between proteins, genes, and small molecules.

Yakushiji et al. [115] propose a method based on full parsing with a large-scale, general-purpose grammar. A parser converts the various sentences that describe the same event into a

[7] A directed graph representing dependencies of words in a sentence.

canonical structure (argument structure) regarding the verb representing the event and its arguments such as the (semantic) subject and object. Interaction extraction is done using pattern matching on the canonical structure.

Deep parsing approaches analyze the whole sentence structure to achieve higher accuracy. However dealing with the whole sentence is time-consuming and expensive. Furthermore these approaches cannot process all kinds of sentences and they usually cannot find the correct structure of complex sentences.

2.3.2 Rule-based methods

In rule-based approaches a set of rules is used. These rules express the format of the text containing relations between entity names. These rules can be defined over words or part-of-speech (POS) tags. The rule-based methods and computational linguistics-based methods are different. Rule-based methods have explicit rules and these rules can be non-linguistic. The BioNLP module [76] is a rule-based module which finds protein names in text and extracts protein-protein interactions using pattern matching. The following five rules are some examples of pattern matching rules in BioNLP. In these rules the symbols A, B refer to protein names and *fn* refers to the interaction verb or its nominalized form.

- *A* ... *fn* ... *B*: This rule models basic sentences like “A inhibits B, C, and D”;
- *A* ... *fn* of ... *B*: This rule models sentences such as “A, an activator of B, is found to be lacking in the patient population”.;
- *A* ... *fn* by ... *B*: This rule describes sentences in the passive voice, such as “A is inhibited by the activities of B.”;
- *A* ..., which ... *fn* ... *B*, ...: This rule models sentences such as “A, which inhibits the activities of B, is found to be lacking in the patient population”;
- *fn* of *A* is ... *B*: This template models sentences such as “Induction of A is caused by B.”.

Obviously, such simple rules cannot produce good results. Many people have used some rules that they produced manually. Using predefined rules can generate nice results but it is time-consuming and if you want to move to another domain, significant manual work is required to change the rules appropriate for the new domain. So, some people tried to automatically construct the entity interaction patterns.

Huang et al. [49] propose a method based on dynamic programming [24] to discover patterns to extract protein interactions. As a first step, part-of-speech tagging is used. Then using dynamic programming by processing and aligning sentences, similar parts in the sentences could be extracted as patterns. They use a threshold d in the algorithm. A pattern is discarded if it appears fewer than d times. Using these patterns, protein–protein interactions can be identified.

Fundel et al. [38] describe a system called RelEx. RelEx generates dependency parse trees using the Stanford Lexicalized Parser[8]. It extracts protein–protein interactions in three steps: preprocessing, extracting and post processing. In the preprocessing step, the sentences are POS tagged and then noun-phrase chunks are identified by fnTBL3[9]. The POS-tagged sentences are submitted to the Stanford Lexicalized Parser in order to get a dependency parse tree (Figure 2.4, upper panel) for each sentence and assign word positions to each word. Gene and protein names are identified by ProMiner [44] based on matching to a synonym dictionary. A dependency chunk tree is produced. In this tree, for each chunk, the corresponding nodes in the dependency tree are combined into a chunk-node (Figure 2.4, lower panel).

In the extracting phase, the candidate relations are extracted from dependency parse trees. To find these relations, some paths that connect two protein names are recognized. These paths should contain relevant terms signalling the relation between two proteins. Currently they use three rules to describe protein-protein relations: effector–relation–effectee, relation–of–effectee-by-effector, and relation-between-effector–and–effectee.

[8] http://nlp.stanford.edu/software/lex-parser.shtml

[9] fnTBL is a customizable, portable and free source machine-learning toolkit primarily oriented towards Natural Language-releated tasks. http://nlp.cs.jhu.edu/~rflorian/fntbl/

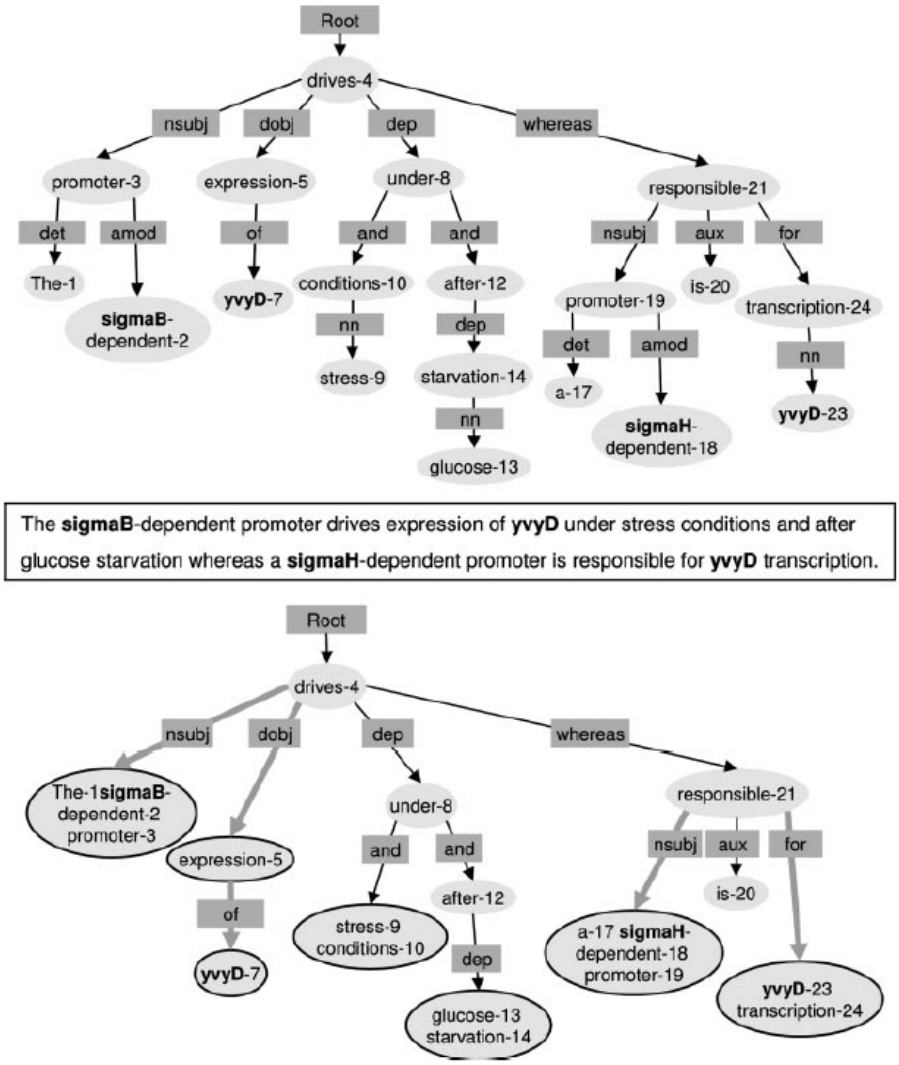

Figure 2.4: Upper panel: Dependency parse tree in RelEx, showing words along with their positions, dependencies, dependency types and the head of the sentence (Root). Lower panel: Corresponding chunk dependency tree [38].

For example in the first rule, if a chunk is recognized as the subject it is marked as the start point and RelEx will search for a relation and a protein or gene name as the end point. If the dependency tree does not contain any subject each protein or gene name can be a start point or an end point and the path between each two names can be their relation.

In the post-processing step candidate relations are filtered. Negative relations are omitted, because they show that there is no relation between proteins. If a node or one of its child nodes contains negative words such as no, not, nor, neither, without, lack, fail(s,ed), unable(s), abrogate(s,d), absen(ce,t), the relation is considered negative.

Effector–effectee position may change. Usually the first name in a sentence is the effector and the second one is considered the effectee. But in some situations like passive sentences, the effector and effectee places are interchanged.

This approach supports enumeration. This can be detected from the dependency tree. If noun phrase chunks are connected by the relations *and*, *or*, *nn* [10], *det* [11] or *dep* [12], each chunk is analyzed separately and for each of them a separate relation is created.

A list of terms is created. This list shows more important relation words and contains interaction verbs and derived nouns and adjectives. Involved terms in candidate relations are checked against this list.

SemGen [88] uses a mechanism for interpreting semantic relationships based on dependency grammar rules to identify and extract semantic propositions on the causal interactions of genes and diseases from biomedical literature.

It has been found that the ability of rule-based approaches to recognize interactions is limited. Rules are not complete enough to cover all situations. Rule-based approaches can only process short and straightforward statements correctly. However biomedical texts are usually complex. Furthermore, they cannot recognize some important features of sentence construction such as mood, modality, and sometimes negation, which can change the meaning of sentences

[10] A noun compound modifier of an NP is any noun that serves to modify the head noun.

[11] A determiner is the relation between the head of an NP and its determiner.

[12] A dependency is labeled as *dep* when the system is unable to determine a more precise dependency relation between two words.

[118].

Ibn Faiz's rule-based approach [34]

Ibn Faiz [34] proposed a rule-based method for extracting protein-protein interactions. This approach is an extension of RelEx [38]. In this method the dependency tree of each sentence is made. The tree is traversed according to a set of rules and various candidate dependency paths are extracted. Each candidate path undergoes a filtering stage. The author prepared a list of interaction terms by combining relation lists used in previous works by [13] and [38], which indicate the occurrence of a relationship in a sentence. If a candidate path does not contain any relation term then this path is removed from further consideration.

This method is able to find relationships with the following patterns:

- PROTEIN1 Relation PROTEIN2
- Relations in which the entities are connected by one or more prepositions:
  - PROTEIN1 *Relation (of | by | to | on | for | in | through | with)* PROTEIN2; e.g. PROTEIN1 *binding by* PROTEIN2 *is blocked by MAb.*
  - *(PREP | REL | N)+(PREP)(REL | PREP | N)** PROTEIN1 *(REL | N | PREP | PROT)+* PROTEIN2; where PREP is any preposition, REL is any relation term, N is any noun, and PROT is any protein instance. e.g. *Activation of* PROTEIN1 *by* PROTEIN2 *in NIH 3T3 cells and in vitro.*
  - *Relation (of | by | to | on | for | in | through | with | between)* PROTEIN1 and PROTEIN2, e.g. *A direct interaction between* PROTEIN1 *subunits and* PROTEIN2.
- PROTEIN1 (/ | \ | −) PROTEIN2; for extracting relations of the form "PROTEIN1/PROTEIN2" binding or "PROTEIN1-PROTEIN2" compound.

This method has been evaluated on the unified PPI corpora [84] and the results are illustrated in Table 2.3.2. These results are at least as good as results obtained by a reproduction

Corpus	AIMed	BioInfer	HPRD50	IEPA	LLL
Precision	44.85	55.82	72.0	68.02	78.40
Recall	60.9	39.15	66.26	69.85	77.44
F-score	51.65	44.97	69.01	68.92	77.91

Table 2.1: Ibn Faiz's rule-based PPI method performance.

of RelEx [84] on the corpora used in the evaluation and this system achieved better results on AIMed and BioInfer.

2.3.3 Machine learning and statistical methods

A number of different machine learning (ML) methods have been proposed ranging from simple methods such as deducing a relationship between two terms based on their co-occurrence in suitable text fragments to more complex methods which employ NLP technologies.

Katrenko and Adriaans [54] propose a representation based on dependency trees which takes into account the syntactic information and allows for using different machine learning methods.

Craven [26] describes two learning methods (Naïve Bayes and relational learning) to find the relations between proteins and sub-cellular structures in which they are found. The naïve Bayes method is based on statistics of co-occurrence of words. To apply the relational learning algorithm, text is first parsed using a shallow parser.

Marcotte et al. [67] describe a Bayesian approach to classify articles based on 80 discriminating words, and to sort them according to their relevance to protein-protein interactions.

Bui et al. [13] propose a hybrid method for extracting protein-protein interactions. This method uses a set of rules to filter out some PPI pairs. Then the remaining pairs go through a SVM classifier.

Stephens et al. [102], Stapley and Benoit [101], and Jenssen et al. [52] discuss extracting the relation between pairs of proteins using probability scores.

Corpus	AIMed	BioInfer	HPRD50	IEPA	LLL
Precision	73.69	82.35	79.33	79.39	98.0
Recall	59.61	73.90	77.34	75.75	87.66
F-score	65.85	77.86	77.52	77.27	88.12

Table 2.2: Ibn Faiz's machine learning-based PPI method performance.

Ibn Faiz's machine learning-based approach [34]

Ibn Faiz [34] proposed a machine learning approach to extract protein-protein interactions. This method considers this problem as a binary classification task. Ibn Faiz uses a maximum entropy classifier. The Stanford dependency parser produces a dependency tree for each sentence. For each pair of proteins in a sentence the dependency path between them, the parse tree of the sentence and other features are extracted. These features include: dependency features coming from the dependency representation of each sentence, syntactic features and surface features derived directly from the raw text (the relation terms and their relative position).

The extracted features along with the existence of a relationship between protein pairs in a sentence make a feature vector. A machine learning model is trained based on these positive (interacting) and negative (non-interacting) pairs of proteins. To avoid sparsity and overfitting problems, feature selection has been done by Mallet (see Chapter 5).

Table 2.3.3 represents the 10-fold cross validation results obtained by testing this system on the unified PPI corpus [84]. These results are competitive with results reported by a hybrid protein-protein interaction method [13].

2.4 Conclusion

In this chapter we discussed a general relation extraction system and the necessary modules for implementing such a system. Each module was explained separately and the available methods and the previous works related to it were described. The only module left is the *Visualization*

[32] which provides a friendly interface for users to acceess the extracted information and it is beyond the scope of this thesis.

Chapter 3

Semi–Supervised Machine Learning

Training classifiers using supervised machine learning requires a large number of feature values—class label pairs. Obtaining this labelled data for training machine learning classifiers is not always easy. Annotating the data is a time-consuming task. For applications in some specific domains, annotation must be done by experts and it can be expensive. As well, sometimes special devices are needed for annotating the data. However, in most cases it is easier and cheaper to obtain unlabelled data than labelled data. Semi-supervised learning is a bootstrapping method which incorporates a large amount of unlabelled data to improve the performance of the supervised learning methods which lack sufficient labelled data.

At first it might seem impossible that unlabelled data can help to improve machine learning systems. Summarizing the explanation given in [119], we try to illustrate the idea behind semi-supervised learning methods. Let each instance be represented by a one-dimensional feature $x \in R$. Assume that we have only two annotated instances, each from a different class: positive and negative. Now we want to find the best boundary between these classes. Figure 3.1 is used to visualize what we now describe. If we only consider the two tagged instances, like what we have in supervised learning, the best estimate of the decision boundary would be x=0. On other hand, if we consider the unlabelled instances shown in green, we observe that they form two separate groups. Assuming that the instances in each class belong to one coherent group ($p(x \mid y)$ is a Gaussian distribution) the best decision boundary between these two classes

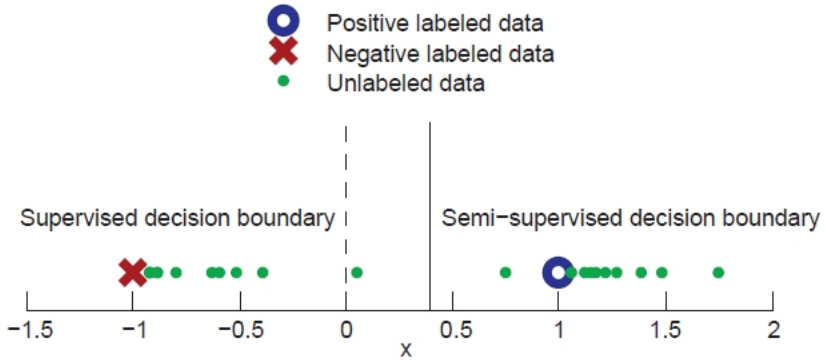

Figure 3.1: How semi-supervised learning works. This Figure is taken from [119].

seems to be $x \approx 0.4$.

As we saw in this example the assumption we take about the relationship between x and $p(x \mid y)$ has an important role in the success of the semi-supervised learning method that we choose. There are different types of semi-supervised learning methods; co-training, self-training, generative models, graph-based learning and so on. The difference between these semi-supervised learning methods lies in the difference between the assumptions they make about the relationship between x and $p(x \mid y)$. Making a wrong assumption leads us to choose a wrong method and ends up generating worse results than using only the labelled data. Figure 3.2 represents an example to clarify this point.

Assume that we have a classification task with two different classes. The top panel of Figure 3.2 shows two Gaussian distributions corresponding to these two classes. These distributions heavily overlap. The dotted line in the middle of the two distributions is the real decision boundary. The other panels in the figure show the decision boundaries found by four different algorithms having five different sets of labelled and unlabelled data.

A supervised learning algorithm only considers the labelled instances and ignores the unlabelled instances. It draws a decision boundary at the mid-point between the negative labelled instance and the positive labelled instance. Whenever we change the labelled instances this decision boundary changes and it is always off a little bit. Supervised learning decision boundary has high variance.

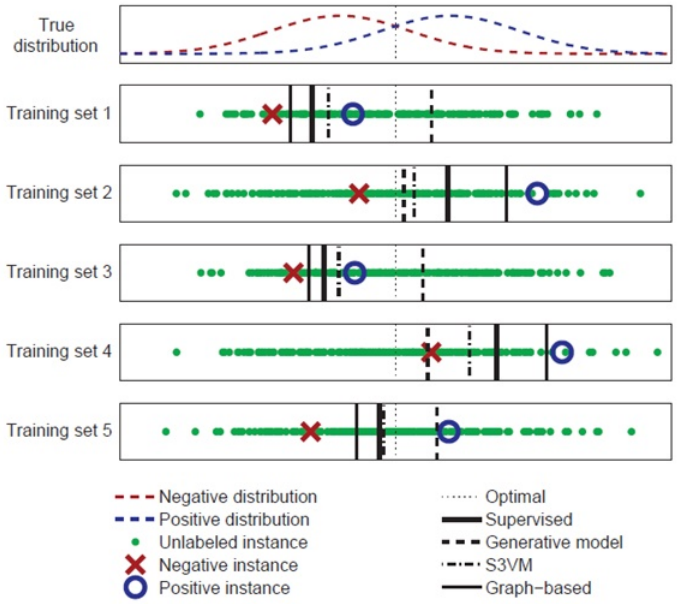

Figure 3.2: Two classes drawn from overlapping Gaussian distributions (top panel). Decision boundaries learned by several algorithms are shown for five random samples of labelled and unlabelled training samples. This figure is taken from [119].

A generative model is a semi-supervised learning model which assumes two classes have two Gaussian models and finds these distributions using the expectationmaximization (EM) algorithm [75]. This method makes the correct assumptions so its decision boundaries are close to the correct decision boundary and also close to one another i.e. , this method has low variance.

S3VM is a semi-supervised Support Vector Machine, which assumes that the decision boundary should not pass through dense unlabelled data regions. This is not always a correct assumption especially in the instances shown here. However the result decision boundary does not seem so bad because this approach has also used the fact that the two classes have approximately the same number of instances.

Graph-based semi-supervised learning has a special way of generating the graph. In this graph any two instances of labelled and unlabelled data are connected by an edge. The edge weight depends on how close (large weight) or far away (small weight) the two instances are. The model assumption is that the instances connected with large weight edges have the same label. However in this particular example where two distributions overlap, instances from two different classes can be quite close yet be connected with large weight edges. So the model prediction is not close to the correct distributions of the classes and this model performs even worse than the supervised learning method.

As the above example shows, making a correct assumption is very important in semi-supervised learning. It is important to choose a suitable semi-supervised algorithm for the problem. But how to choose this algorithm still is an open question.

In the remainder of this chapter we will discuss two important semi-supervised learning algorithms: co-training and self-training algorithms.

3.1 Co-training

Co-training [10] is a kind of semi-supervised learning. In co-training we have two (or more) different views of the data (each example is described by two separate feature sets). Starting

with the labelled data, the classifier learns from each view separately. Then, using the two learned models, the unlabelled data gets annotated and the most confident predictions from each model are added to the labelled data. This process continues for a number of iterations. As each classifier is providing extra, informative labelled data for the other classifier(s) the training should progress.

In the original definition of co-training, Blum and Mitchell [10] made two important assumptions about the effectiveness and applicability of co-training: (1) each of the two views is sufficient to classify the data; (2) the two views are conditionally independent given the class label. When these two assumptions hold, Blum and Mitchell [10], derive PAC-like[1] guarantees on learning. However, Abney [4] argues that the Blum and Mitchell independence assumption is too restrictive and typically violated in the data, and shows a weaker independence assumption suffices. He also proposes a greedy algorithm to maximize agreement on unlabelled data between classifiers, which produces good results in a co-training experiment for named entity classification.

Moreover, Clark et al. [19] investigate the use of co-training in POS tagging. They use two types of co-training. In the first one (agreement-based co-training), theoretical arguments of Abney [4] that directly maximize the agreement rates between the two classifiers are used. In the second one a naïve co-training process which simply retrains classifiers on all the newly labelled data is tested. The results show that the naïve co-training process leads to similar performance, at a much lower computational cost.

Later research by Nigam and Ghani [78] shows that when the feature set is naturally divided into independent and redundant splits, co-training algorithms outperform other algorithms using unlabelled data.

In natural language processing, co-training has been used for word sense disambiguation [74], statistical parsing [92], reference resolution [77], part-of-speech tagging [19], etc. and has achieved good results when lacking an annotated corpus.

[1] Probably Approximately Correct (PAC) is a framework for a mathematical analysis of machine learning.

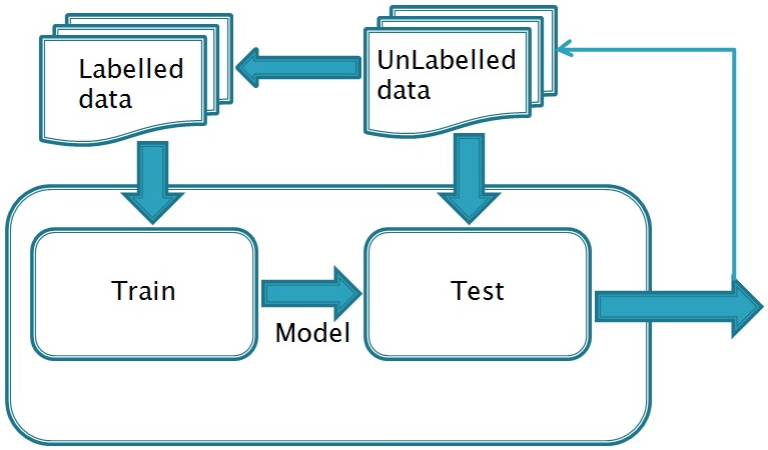

Figure 3.3: Self-training algorithm

3.2 Self-training

Self-training is another type of semi-supervised learning. While there is a common agreement about the co-training definition, different definitions have been proposed for self-training.

Ng and Cardie [77] define self-training as "a single-view weakly supervised learning algorithm" and show that it outperforms the multi-view algorithms when there are not obvious independent and redundant feature splits. They used bagging [12] and majority voting [66] in their implementation. A set of classifiers get trained on the labelled data then they classify the unlabelled data independently. Only those predictions which have the same label by all classifiers are added to the training set and the classifiers get trained again. This process continues until a stop condition is met.

Clark et al. [19] define self-training differently. In self-training "a tagger is simply retrained on its own labelled cache at each round". This definition is in agreement with Nigam and Ghani's [78] definition. We adopt this definition in our work. Figure 3.3 illustrates the self-training algorithm according to this definition. Based on this definition there is only one classifier which is trained on labelled data, then the resulting model is used to train the unlabelled data, the most confident predictions are added to the training set and the classifier is retrained on this new training set. This procedure repeats for several rounds.

Self-training assumes that its own predictions, at least the high confidence ones, are correct. This is a correct assumption if the classes are well-separated clusters.

It is understandable that early mistakes in the first iterations of the algorithm can reinforce themselves through the next iterations by generating incorrect labels. So re-training with this data can cause worse models in each iteration. Various heuristics have been proposed to solve this problem.

The major advantage of the self-training algorithm is that it is a wrapper method and it is very simple to apply. If given very complicated learning algorithm, it can be used within the self-training framework as a black-box, without needing to know its details.

Yarowsky [116] applied self-training learning for word sense disambiguation. He used the "one sense per collocation" and the "one sense per discourse" properties of human languages for word sense disambiguation. From observation, words tend to exhibit only one sense in most given discourse and in a given collocation.

Self-training has been applied for learning subjective nouns [87], spam filtering [23] and phenotype name recognition [56].

Chapter 4

Conditional Random Fields

Segmenting and labelling sequences is fundamental in many different applications in several scientific fields including bioinformatics, computational linguistics and speech recognition [33, 69, 85]. Part-of-speech tagging is one simple example of such an application.

Hidden Markov Models (HMMs) [86] are commonly used techniques in labelling sequence data. HMMs are a form of generative model that define the joint probability of $p(X, Y)$ where X is a random variable over observation sequences and Y is a random variable over corresponding label sequences.

$$p(X, Y) = p(X)p(Y|X) \tag{4.1}$$

For example, in part of speech tagging, X ranges over tokens in a sentence and Y ranges over the POS tags. According to HMMs independence assumption the current observation is statistically independent of the previous observations. To find the joint probability over observation and label sequences a generative model has to enumerate all possible observation sequences which is not possible unless observation elements are represented as isolated units such as words or nucleotides, independent from the other elements in an observation sequence. Practically the dimensionality of observations (X) can be very large and the features may also have complex dependencies, so making a probability distribution over them is difficult and the inference problem for such models is intractable [61].

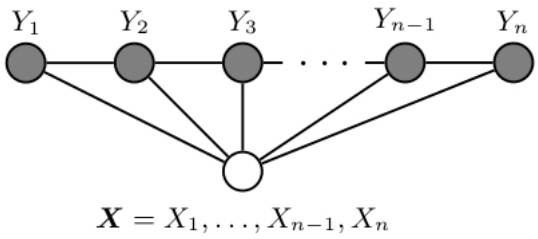

Figure 4.1: Graphical structure of a chain-structured CRF for sequences. This is taken from [111].

An alternative representation method is the Conditional model. A conditional model specifies the conditional probability $p(Y \mid X)$ directly. Therefore there is no need to model the observation and also to make the independence assumption between features.

Conditional Random Fields (CRF) is a non-generative probabilistic framework for labelling and segmenting sequential data based on conditional models. The conditional nature of CRF relaxes the independence assumptions required in HMMs, resulting in enriched features and improved performance.

A conditional random field can be considered as an undirected graph or Markov random fields [20], conditioned on X. Let $G = (V, E)$ be an undirected graph such that there is a node $v \in V$ for each $Y_v \in Y$. Then $G = (V, E)$ is a conditional random field if each random variable Y_v obeys the Markov property [1] with respect to the graph. Theoretically the structure of graph G may be arbitrary but the simplest and most common graph structure in CRF is shown in Figure 4.1 in which the elements corresponding to elements of Y form a first-order chain.

According to the fundamental theorem of random fields [43], we can factor the joint distribution over the label sequence Y given X based on the cliques in G [61].

$$p(Y|X) = \exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)) \tag{4.2}$$

where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of the entire observation sequence and the

[1] A stochastic process has the Markov property if the conditional probability distribution of future states of the process (conditional on both past and present values) depends only upon the present state.

labels at positions i and $i-1$ in the label sequence; $s_k(y_i, x, i))$ is a state feature function of the label at position i and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data.

We assume that feature functions t_j and s_k are given and fixed. For example a boolean feature s_k is true if the word X_i is uppercase and tag Y_i is "proper noun".

The parameter estimation problem is to determine the parameters λ_j and μ_k from training data. Lafferty et al. [61] proposed two iterative scaling algorithms to find the parameters that maximize the log-likelihood of the training data.

CRFs have been used successfully in many applications. In the NLP domain CRFs have been applied for named entity recognition (NER) [56, 62, 97], shallow parsing [97, 104], segmenting addresses in Web pages [27], information integration [112], finding semantic roles in text [91], identifying the sources of opinions [18], word alignment in machine translation [11], citation extraction from research papers [82], extraction of information from tables in text documents [83], Chinese word segmentation [81], Japanese morphological analysis [60], and many others.

BANNER [62] is a biomedical NER system which is based on CRF. This system has been used in this research project for extracting genotype names. As CRF id the best method for sequence tagging, we proposed a CRF-based machine learning method for extracting phenotype names (see Chapter 8).

Chapter 5

Natural Language Processing Tools

We used several available NLP tools to implement our proposed methods. This Chapter gives a short description for each tool that has been used in this thesis.

5.1 MetaMap

MetaMap, a program developed by the National Library of Medicine (NLM) [7], provides a link between biomedical text and the structured knowledge in the Unified Medical Language System (UMLS) Metathesaurus. To map phrases in the text to concepts in the UMLS Metathesaurus, MetaMap analyzes the input text lexically and semantically. First, MetaMap tokenizes the input text. In the tokenization process the input text is broken into meaningful elements, like words. After part-of-speech tagging and shallow parsing using the SPECIALIST Lexicon [1], the text has been broken into phrases. Phrases then undergo further analysis: Each phrase is mapped to a set of candidate UMLS concepts, each candidate being given a score that represents how well the phrase matches the candidates. An optional last step is word sense disambiguation (WSD) which chooses the best candidate sense with respect to the surrounding text [7].

[1] The SPECIALIST Lexicon contains information about common English vocabulary, biomedical terms, terms found in MEDLINE and terms found in the UMLS Metathesaurus.

```
Phrase: "Fanconi anemia"
>>>>> Syntax
msu
  head([lexmatch([Fanconi anemia]),inputmatch([Fanconi,anemia]),
   tag(noun),tokens([fanconi,anemia])])
<<<<< Syntax
>>>>> Phrase
fanconi anemia
<<<<< Phrase
>>>>> Candidates
Meta Candidates (Total=4; Excluded=1; Pruned=0; Remaining=3)
  1000   Fanconi Anemia [Disease or Syndrome]
   861   Anaemia (Anemia) [Disease or Syndrome]
   861   Anemia (Genus Anemia) [Plant]
   789 E anaemic [Finding]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (1000):
  1000   Fanconi Anemia [Disease or Syndrome]
<<<<< Mappings
```

Figure 5.1: MetaMap output for “the Fanconi anemia”

MetaMap is configurable with options for vocabularies and data models in use, output format and algorithmic computations. An example of the human-readable output format for the text “Fanconi anemia” from the sentence “Fanconi anemia is a genetic disease with an incidence of 1 per 350,000 births.” is shown in Figure 5.1. MetaMap finds 4 candidates for this phrase and after WSD it maps the phrase to the “Disease or Syndrome” concept.

In UMLS each Metathesaurus concept is assigned to at least one semantic type. In Figure 5.1 the semantic type of each concept is given in the square brackets. Semantic types are categorized into groups, called Semantic Groups (SG), that are sub-domains of biomedicine such as *Anatomy*, *Living Beings* and *Disorders* [72]. Each semantic type belongs to exactly one SG.

Figure 5.2 illustrates a part of this hierarchy. In this Figure *Physiology*, *Disorders* and *Anatomy* are semantic groups and the other ones are all semantic types under SG *Disorders*. A list of semantic types and semantic groups used in this project is available in Appendix A.

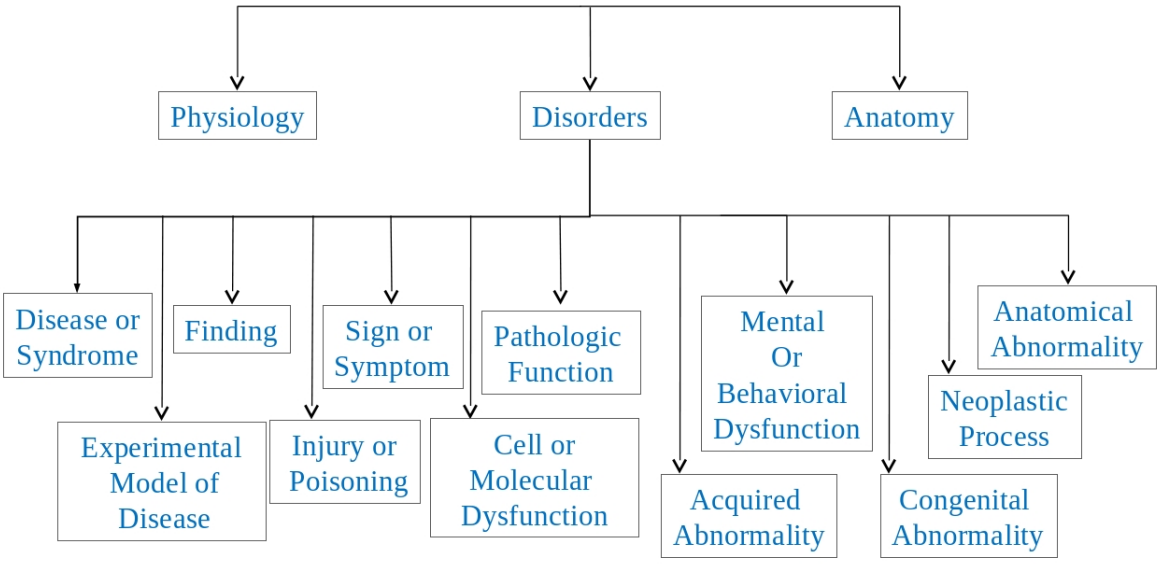

Figure 5.2: A part of UMLS semantic types and semantic groups hierarchy.

5.2 Mallet

MAchine Learning for LanguagE Toolkit (Mallet) [70] is an open source, java-based package for natural language processing, document classification, clustering, topic modelling, information extraction, and other machine learning applications on text. This software has been written by Andrew McCallum with contributions of several graduate students and staff at the University of Massachusetts Amherst. Mallet includes efficient routines for converting text to "features", a wide variety of algorithms (including Naive Bayes, Maximum Entropy, and Decision Trees), and code for evaluating classifier performance using several commonly used metrics. In addition Mallet includes tools for sequence tagging which is useful in a variety of NLP applications such as Named Entity Recognition. Algorithms include Hidden Markov Models, Maximum Entropy Markov Models, and Conditional Random Fields. Mallet is capable of converting the text into a numerical representation for efficient processing. In this project the Maximum Entropy algorithm in Mallet has been used for finding the relationships between phenotypes and genotypes in a sentence. We used version 2.0.7.

5.3 BLLIP reranking parser

The BLLIP reranking parser [17] (also known as the Charniak-Johnson parser, the Charniak parser and the Brown reranking parser) is a statistical parser developed by Charniak and Johnson. This parser is composed of the Charniak parser followed by a reranker. The parser uses a dynamic programming n-best parsing algorithm that utilizes a heuristic coarse-to-fine refinement of parses. The n^2 best parses produced by this algorithm go through a Maximum Entropy discriminative reranker. The reranker selects the best parse among the input parses using a wide variety of features. The original model in BLLIP was trained on the Wall Street Journal articles in Penn TreeBank corpus [68] but McClosky and Charniak [71] used a self-training method to produce a biomedical model.

5.4 BioText

BioText [93] is an open source tool based on a simple algorithm for finding abbreviations and their full forms from biomedical text. Although the algorithm is very simple, it is really effective and reliable. Furthermore it is very fast, which makes it suitable for many NLP tasks.

5.5 PostMed

PostMed is a local modification of MedPost [100] done by Art Bugorski. MedPost is a high accuracy part-of-speech-tagger for biomedical text. MedPost only works on abstracts but its modified version, PostMed, works on full articles. During the process of POS tagging the text is segmented to sentences. In this project we use PostMed to segment the text into sentences because our experiments showed that it is the best for biomedical text.

[2] n is 50 by default.

5.6 Stanford dependency parser

The Stanford dependency parser[3] is a statistical parser that generates the Stanford dependency representation. The Stanford typed dependencies representation provides a simple description of the grammatical relations in a sentence. In this representation every relationship is illustrated by a triple of relations between two words. For example "The subject of ate is John" in sentence "John ate an apple.". These dependencies are mapped to a directed graph representation in which words in the sentence are nodes in the graph and grammatical relations are edge labels [30]. In this project the Stanford dependency parser has been used to provide a dependency representation of sentences because it is the best dependency parser available.

[3] http://nlp.stanford.edu/software/stanford-dependencies.shtml

Chapter 6

Genotype Name Recognition

Extracting gene names from biomedical literature has been extensively studied and many methods have been proposed for this task. We used, BANNER, one of the best available methods in this research project.

BANNER [62] is an open-source biomedical named entity recognition system implemented using second order conditional random fields (CRF), a machine learning technique. The BANNER architecture is illustrated in Figure 6.1. A BANNER input file consists of a text which has been separated into sentences. Each sentence is taken individually and is tokenized. The tokenization process in BANNER breaks tokens into either a contiguous block of letters and/or digits or a single punctuation mark. As an example, the string "Bub2p-dependent" is broken into three tokens: "Bub2p", "-", and "dependent".

In the next step, features are assigned to each individual token. Each feature is a name/value pair for use by the machine learning algorithm. And finally in the labelling process, each feature gets exactly one label. BANNER makes use of the Mallet CRF [70] in both feature generation and labelling. The set of machine learning features used in BANNER is listed in Table 6.1.

BANNER considers a token window of 2 to make features, meaning that the features of each token contain the features of the two previous and the two following tokens. It uses the IOB label model (Inside, Outside, beginning).

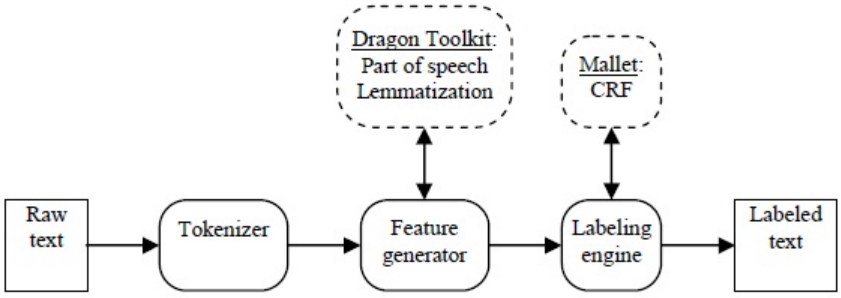

Figure 6.1: BANNER Architecture [62]

Feature set definition	Description
The part of speech which the token plays in the sentence	Provided by the Dragon toolkit implementation of the Hepple tagger.
The lemma for the word represented by the token, if any	Provided by the Dragon toolkit.
A set of regular expression features	Includes variations on capitalization and letter/digit combinations.
2, 3 and 4-character prefixes and suffixes	
2 and 3 character n-grams	Including start-of-token and end-of-token Indicators
Word class Convert	upper-case letters to “A”, lowercase letters to “a”, digits to “0” and other characters to “x”
Numeric normalization	Convert digits to “0”
Roman numerals	
The names of the Greek letters	

Table 6.1: Set of features in BANNER [62]

BANNER detects situations in which the matching of parentheses, brackets or double quotation marks receives different labels. Since these punctuation marks are always paired, they should have the same labels otherwise the labelling engine made a mistake. In the case of unmatched labels BANNER will drop these named entities.

BANNER has been used for NER in the gene names and the disease names domains, and it has achieved results superior to the most commonly used baseline NER systems in the literature (LingPipe [15] and ABNER [96]). Table 6.2 provides a representative performance of BANNER, ABNER and LingPipe in genotype name recognition. The evaluation was performed using the freely available BioCreative 2 GM corpus [1] and 5×2 cross validation. The BioCreative 2 GM corpus contains 15,000 sentences from MEDLINE abstracts and mentions over 18,000 entities. This corpus was prepared for the BioCreative contest in October 2006. We used the same corpus to train BANNER and extract genotypes from text.

Method	Precision	Recall	F-measure
BANNER	85.09	79.06	81.96
ABNER	83.21	73.94	78.30
LingPipe	60.34	70.32	64.95

Table 6.2: BANNER evaluation results [62]

Chapter 7

Rule–Based Phenotype Name Recognition

The last few years have seen a remarkable growth of NER techniques in the biomedical domain. However, these techniques tend to emphasize on extracting genes, proteins, diseases and drugs names. Currently, many systems which use phenotypes to find information like phenotype-genotype relations or build semantic networks (for instance, [25]) use only dictionary-based techniques to recognize the phenotypes in the text.

Many biomedical dictionaries and databases are made for applications in this domain. However we are not aware of any data source which is both comprehensive and ideally suited for phenotype name recognition. Generally, because the speed of introducing new concepts like new phenotypes to biomedicine world is so fast, no data source can keep up with new terms and concepts.

The Unified Medical Language System (UMLS) Metathesaurus [50] is a very large, multi-purpose, and multi-lingual vocabulary database that contains more than 1.8 million concepts. Each concept in UMLS is mapped to at least one semantic type (see examples of semantic types in Appendix A) but *Phenotype* is not one of UMLS semantic types. Therefore, UMLS does not give enough information for recognizing phenotypes in text.

The aim of Pharmacogenetics Knowledge Base (PharmGKB) [57] is to gather the information about the effect of human genetic variation on drug–response phenotypes from published papers. It is a highly respected database queried by clinicians and bioinformaticians. This

manually curated database summarizes published gene–drug–phenotype relationships. Clearly this manual process cannot keep the database as up to date as it should be.

The Online Mendelian Inheritance in Man (OMIM) [73] is an important and comprehensive information source of human genes and genetic phenotypes [90]. However, OMIM does not use a controlled vocabulary to describe the phenotypic features in its clinical synopsis section making it inappropriate for data mining purposes [90]. It was also manually curated.

The Human Phenotype Ontology (HPO) [90] provides a standardized vocabulary of human phenotypes. It has been developed using information from OMIM. Although it contains approximately 10,000 terms, it is incomplete. It is constantly refined, corrected, and expanded manually but it has difficulty keeping pace with all the new phenotypes.

We proposed two different approaches to recognize phenotype names in text. The first one is a rule–based approach [55] and the second one is a machine learning–based system [56]. The rule-based system is explained in this chapter and the machine-learning based system is elaborated in Chapter 8.

7.1 The proposed method

We started working on a phenotype name recognition system when we could not find any available comprehensive resource or method for this application. We started using HPO but as it is not complete, it does not list all published phenotypes. We ended up developing our own system which is able to detect phenotype names even if they are not available in HPO. It is worth noting that phenotypes can be classified into two groups. Phenotypes like *hair color* that correspond to a normal characteristics in human being and phenotypes related to abnormalities. Most phenotypes in the first group are available in HPO. We are more interested in finding the phenotypes in the second group and finally extracting their relationships with phenotypes.

UMLS is another source of data which could be used for extracting phenotypes. The UMLS Semantic Network includes 133 semantic types. Although *Phenotype* is not one of these semantic types, they still give important and useful information for detecting phenotypes. These

semantic types are categorized into 15 Semantic Groups (SG) [72] that are more general and more comprehensive for non-experts. The Semantic Group *Disorders* is one of these semantic groups which contains semantic types very close to the definition of phenotype. This semantic group has been used in other research [14] to map terminologies between the Mammalian Phenotype Ontology (MPO)[99] and the Online Mendelian Inheritance in Man (OMIM) [73]. The Semantic Group *Disorders* contains the following semantic types: *Acquired Abnormality, Anatomical Abnormality, Cell or Molecular Dysfunction, Congenital Abnormality, Disease or Syndrome, Experimental Model of Disease, Finding, Injury or Poisoning, Mental or Behavioural Dysfunction, Neoplastic Process, Pathologic Function, Sign or Symptom*.

In our first attempt to develop such a system, we integrated the valuable information in both HPO and UMLs Metathesaurus. In this method, the text was processed by MetaMap and each noun phrase was mapped to a semantic type. If the semantic type was in SG *Disorders* we considered it as a phenotype. However, after testing this method and analyzing positive and negative results, a more sophisticated system was proposed. This system incorporates the available knowledge in UMLs Metathesaurus and HPO and integrates it with five rules to enhance the system and improve the results. A block diagram representing our system processing is shown in Figure 7.1. The system performs the following steps:

I MetaMap (see Chapter 5) tokenizes the input text and finds the boundary of sentences and then chunks each sentence into phrases. Each noun phrase is assigned to several UMLS semantic types and finally after word sense disambiguation, the best semantic type(s) is provided. We used the strict model and word sense disambiguation embedded in MetaMap.

II The Disorder Recognizer uses the MetaMap output to find phenotypes and phenotype candidates. This part is original to our system and is described in detail in Section 7.1.1.

III OBO-Edit [29] is an open source tool that makes it possible to edit or search in ontology files in OBO format. The phenotype candidates found by Disorder Recognizer are searched in HPO. The candidates available in the HPO are considered as phenotypes.

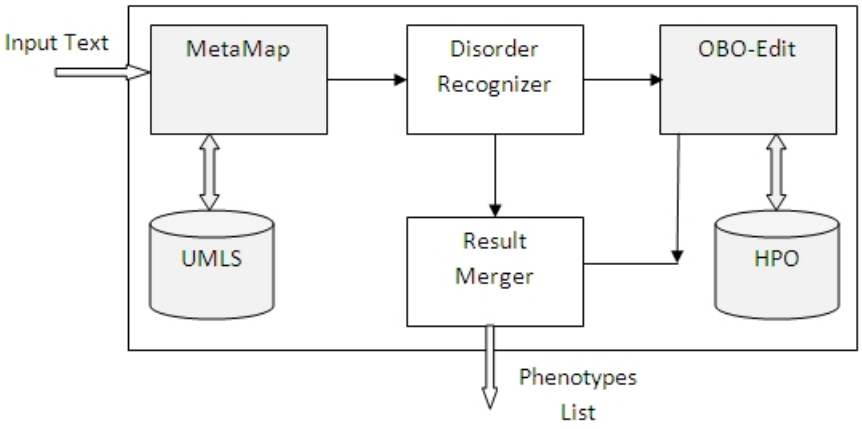

Figure 7.1: System block diagram.

IV Result Merger merges the phenotypes found by Disorder Recognizer and OBO-Edit and makes the final list of phenotypes extracted from the input text.

7.1.1 Disorder recognizer

The text was processed by MetaMap and noun phrases were matched by semantic types. Now it is the time to extract phenotypes. Our initial plan was to tag every noun phrase with a semantic type in the SG *Disorders* as a phenotype but this approach made some mistakes. After analyzing these mistakes, several post processing steps and five rules were proposed to overcome the remaining problems:

1. A number of problems caused by acronyms. Authors tend to use acronyms in their papers. Unfortunately when phenotypes appear in their short form, it gets difficult for the system to recognize them. MetaMap is able to recognize acronyms references but its database does not contain all the acronyms. Furthermore, some acronyms refer to more than one concept and MetaMap cannot disambiguate their references correctly. Usually the authors indicate the local unambiguous reference for each acronym in its first appearance in the paper. We used this information to create a local list of acronyms and their references in each paper using BioText [93]. This list is used to resolve the

acronyms found in the remainder of the text. The first rule is proposed to solve this problem.

Rule 1 Resolve the acronym referencing problem by making and using a list of acronyms occurring in each paper.

2. In some cases the phenotype contains more than one biomedical or clinical term and the complete phenotype is not available in UMLS. MetaMap breaks these kind of phenotypes into several parts and assigns a different ST to each part. The question is which of these STs should be considered as the semantic type of the whole phrase which helps us to detect the phenotype. Figure 7.2 represents the UMLS output for *"[The] presented learning disabilities"*. There are two separate concepts in the MetaMap output. The first one is *"presented"* which is assigned to the semantic type *[Idea or Concept]* and the second one is *"learning disabilities"* with the semantic type *[Mental or Behavioral Dysfunction]*. As *"presented"* is only an adjective for *"learning disabilities"* so the ST of the whole phrase should be *[Mental or Behavioral Dysfunction]* which is in SG *Disorders*. So, in these situations the semantic type of the noun phrase head is the most important part and our system should consider the head's semantic type in order to recognize the semantic type of the whole phrase. The second rule handles such situations.

   Rule 2 The semantic type of a noun phrase is the semantic type assigned by MetaMap to its head.

3. We found a common template among some phenotypes like *"large ventricles"* that are not recognized by MetaMap. These phenotypes begin with a special modifier followed by terms which are assigned to SGs *Anatomy* or *Physiology* by MetaMap. Burgun et al. [14] mentioned this class of phenotypes where a list of 100 special modifiers (see Appendix B), having to do with some sort of unusual aspect or dysfunction (like *large*, *defective* and *abnormal*), is given. This list was created collecting the modifiers that occur most frequently with MPO terms. We found this list incomplete for our application and three more modifiers found in our small corpus were added to the list. *Missing*,

malformed, and *underdeveloped* are these three modifiers. We may find more modifiers to add to this list in the future. The third rule describes the template of this kind of modifiers.

Rule 3 If a phrase is "modifier (from the list of special modifiers) + [Anatomy] or [Physiology]" it is a phenotype name.

4. In some cases non-phenotype phrases are assigned to some specific semantic types in SG *Disorders* leading to false positives in phenotype name recognition. Analyzing these specific semantic types, we concluded that only assignment of a phrase to one of these semantic types is not enough to consider that phrase as a phenotype. For example, MetaMap assigns *"responsible"* to the semantic type *[Finding]*. The word *"responsible"* is clearly not a phenotype. On the other hand *"overgrowth"*, which is a phenotype, is assigned to the semantic type *[Finding]*, too. The problematic semantic groups are: *Finding*, *Disease or Syndrome*, *Experimental Model of Disease*, *Injury or Poisoning*, *Sign or Symptom*, *Pathologic Function*, and *Cell or Molecular Dysfunction*. If any of these semantic types are mapped to a phrase, that phrase is considered as a phenotype candidate that needs further analysis. Phenotype candidates are searched in HPO in step III of the system process described above to be confirmed as phenotypes. If a phenotype candidate is found in HPO, it is recognized as a phenotype. While making the candidates list we should consider rules 4 and 5 below.

5. In several cases the phenotype is a plural term but only its singular form is available in HPO. One example is *"deep set eyes"*. It is not in the HPO but *"deep set eye"* is. *Rule 4* allows us to consider a plural term as a phenotype, if the singular form is found in HPO.

   Rule 4 If the single form of a phrase is a phenotype the plural form is a phenotype, too.

6. Several phenotypes consist of adjectives and adverbs followed by terms found in HPO. In these cases the whole phenotype phrase cannot be found in HPO, however their head is in HPO. Therefore system will remove the adjective and adverbs and search only for

the heads in HPO.

Rule 5 If the head of a phenotype candidate phrase is a phenotype, the whole phrase is a phenotype.

In summary, the system analyzes all noun phrases extracted by MetaMap and their semantic types, one by one. If a phrase contains an acronym, its reference is resolved based on *Rule 1*. If the phrase matches *Rule 3*, it is added to the phenotype list, otherwise if the phrase is broken into more than one concept and mapped to more than one semantic type, its semantic type is assigned according to *Rule 2*. If the semantic type is in the Semantic Group *Disorder*, the phrase is recognized as either a phenotype or a phenotype candidate. Phenotype candidates are added to the phenotype candidate list along with their heads and their singular form if they are plural (according to Rules 4 and 5), to be processed in step III.

7.2 Evaluation

The proposed Method was evaluated using a corpus containing 120 sentences with 110 phenotype phrases. These sentences were collected from four random full-text journal articles specialized in human genetics. Some sentences do not have any phenotypes. Precision, recall and F-measure were used to measure the performance of this system. Table 7.1 represents the performance of differently configured systems to measure the contributions of each rule in the final results. The basic form is the integration of UMLS and HPO using none of the rules discussed above. The results of adding each of the rules are listed in the table.

As these rules were found from the same corpus, we thought it is better to evaluate the system using a separate test set. This test set contained 96 sentences and 74 phenotypes. This test set was consisted of two randomly selected biomedical papers. Table 7.2 shows the results obtained from testing the system. As the table shows, the results dropped. This is the problem of rule-based methods which cannot achieve their previously good results when they are evaluated on new test sets.

```
Phrase: "[The] presented learning disabilities"
>>>>> Phrase
presented learning disabilities
<<<<< Phrase
>>>>> Candidates
Meta Candidates (9):
   901 Learning Disabilities [Mental or Behavioral Dysfunction]
   882 Learning disability (Learning disability - speciality)
       [Biomedical Occupation or Discipline]
   827 Learning [Mental Process]
   827 Disabilities (Disability) [Finding]
   743 Disabled (Disabled Persons) [Patient or Disabled Group]
   743 Disabled [Qualitative Concept]
   660 Presented (Presentation) [Idea or Concept]
   627 Present [Quantitative Concept]
   627 Present (Present (Time point or interval)) [Temporal Concept]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (901):
   660 Presented (Presentation) [Idea or Concept]
   901 Learning Disabilities [Mental or Behavioral Dysfunction]
<<<<< Mappings
```

Figure 7.2: An example of Rule 1

Method	Precision	Recall	F-measure
Basic Form	88.78	74.21	80.84
Applying Only Rule 1	89.38	78.9	83.81
Applying Only Rule 2	97.19	75.91	85.24
Applying Only Rule 3	89.09	76.56	82.35
Applying Only Rule 4	88.9	75.78	81.32
Applying Only Rule 5	89.38	78.9	83.81
Applying All Rules	97.58	88.32	92.71

Table 7.1: Results of evaluating the system on the original corpus.

	Precision	Recall	F-measure
Applying All Rules	94.87	62.71	75.51

Table 7.2: Results of evaluating the system on a separate test set.

The analysis of errors has shown that some errors originate in shortcomings of our method, but other errors are caused by incorrect information provided by the systems that we use. Some phenotypes are detected as phenotype candidates and searched in HPO but because HPO is not complete, they are not found and as a result they are not recognized as phenotypes. In other cases MetaMap parser breaks a phenotype phrase into two different noun phrases by mistake and it prevents the system from recognizing the phenotype. And finally in several examples, MetaMap makes mistake during WSD. The percentage of total errors that these three sources cause are shown in the Table 7.3.

7.3 Summary

In this chapter we introduced a rule-enhanced dictionary-based phenotype name recognition system. This system integrates the available knowledge in HPO and UMLS Metathesaurus and uses MetaMap in an innovative way to find phenotype names in biomedical texts. Our approach applies five specific rules to enhance the phenotype name recognition procedure.

The performance of the system was evaluated using two different corpora, the one which is used to extract the rules and a separate test set. Results showed that the performance of the system was better for the original corpus. We should keep this in mind that generally, rule-based systems have this problem. Applying the rules extracted from one set to another corpus might reduce the results.

After analyzing the errors, we found that some errors were caused because of HPO incompleteness, Word Sense Disambiguation Function in MetaMap or MetaMap parser.

Collier et al. [21] have combined our proposed rule-based system with machine learning techniques (CRF and HMM) to find phenotype candidates in genetic texts. Their system consists of two modules: the machine learning labeller and the knowledge-based labeller (our rule-based system and dictionary matching) and these modules' predictions are merged. They have considered both cellular-level and supercellular-level phenotypes in their system. They used our manually created corpus to test their system against the rule-based method and it did not outperform our rule-based system. However it had better results on their corpus [21]. Their corpus is called Phenominer and we have used it (see Chapter 10) for extracting genotype-phenotype relationships.

Cause of Error	**Example of Error**	**Description of Error**	**Percentage**
MetaMap parser	Partial hypoplasia of the corpus callosum	MetaMap finds two separate phrases: "Partial hypoplasia"; "of the corpus callosum"	20
	missing vertebrae	MetaMap finds two separate phrases: "missing"; "vertebrae"	
MetaMap WSD	learning deficit	[Functional Concept] chosen instead of [Disease or Syndrome]	25
	triphalangeal thumb	[Gene or Genome] chosen instead of [Congenital Abnormality]	
	aplastic anemia	[Gene or Genome] chosen instead of [Disease or Syndrome]	
	osteosarcoma	[Gene or Genome] chosen instead of [Neoplastic Process]	
	diabetes insipidus	[Functional Concept] chosen instead of [Disease or Syndrome]	
Phenotype candidates not in HPO	thumb duplication		25
	thrombopenia		
	increased erythrocyte adenosine deaminase activity		
	macrocytosis		

Table 7.3: Three sources of errors

Chapter 8

Machine Learning–Based Phenotype Name Recognition

Chapter 7 described our first proposed system for extracting phenotype names from biomedical text. It is a rule-based system which integrates available knowledge in the Human Phenotype Ontology (HPO) [90] and the Unified Medical Language System (UMLS) Metathesaurus [50]. This system achieves good results; however it, like other rule-based methods, has some shortcomings. As we extracted the rules from a small corpus, they may be overfitted to that corpus and cannot cover new phenotypes in other texts. And if we try to use it on a different corpus, we may need to add extra rules. It is not easy to analyze the errors manually to generate the new rules. So, we decided to take a different path and extend the capabilities of our rule-based system using machine learning methods.

BANNER [62] (see Chapter 6) is a biomedical named entity recognition system implemented using second order conditional random fields (CRF) (Chapter 4), a machine learning technique. It has been used for gene and protein and also for disease name recognition. It is open-source and provides a good infrastructure for NER. Its results are convincing and features can be easily added. Therefore starting with BANNER and its CRF method, our rule-based method was the incorporated.

Figure 8.1 illustrates the block diagram of the proposed machine learning-based system.

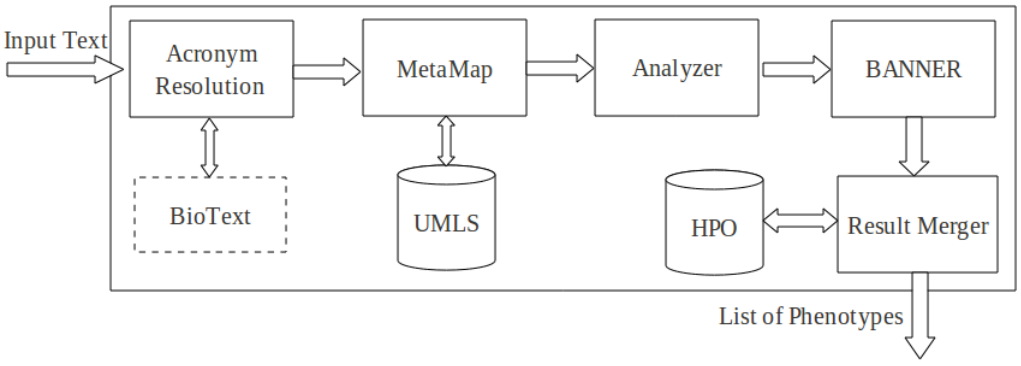

Figure 8.1: System Block Diagram

The first phase in this system is finding acronyms in the input text and resolving them. Usually, papers indicate the local unambiguous reference for each acronym used at its first usage. So a list of local full forms for acronyms is made for each paper and acronym resolution is done based on this list (according to rule 1 of the rule-based method (Chapter 7)). Then the output which does not contain any acronyms is processed by MetaMap. According to our settings, MetaMap finds composite noun phrases with up to three simple phrases and prints out the syntax of each noun phrase. A semantic type is assigned to each noun phrase; these semantic types provide important information for detecting phenotype names. The Analyzer analyzes the MetaMap output and changes it to BANNER input format. Our feature-enhanced BANNER takes each sentence separately and using the features, finds some but not all phenotypes. In the last step the system uses HPO and searches for phenotypes mentioned in HPO in the text. the found phenotypes are added to our list of phenotypes. The details of how we made use of rules and features in our machine learning method are explained in the following sections.

8.0.1 Incorporating the rules

This section explains how we used our rules proposed in the rule-based system in implementing the machine learning-based method.

Rule 1 ***Resolve the acronym referencing problem by making and using a list of acronyms occurring in each paper.***

Rule 1 is implemented in *Acronym Resolution*. *Acronym Resolution* finds the full forms of acronyms using their local unambiguous references in the text and replaces acronyms with their unabbreviated forms. BioText [93] has been used to make a list of acronyms and their full forms in the text.

Rule 2 ***The semantic type of a noun phrase is the semantic type assigned by Metamap to its head.***

Analyzer implements this rule. It finds the noun phrases and their heads processing the MetaMap output. If Metamap breaks a noun phrase into different parts and assigns different semantic types to them, the Analyzer assigns the semantic type of the head to the whole phrase. An example of such a phrase and the Analyzer output is shown in Figure 8.2. The phrase "of Diamond-Blackfan anemia patients" is broken into two noun phrases "Diamond-Blackfan anemia" and "patients", each with a different semantic type. MetaMap output shows that the head of the whole phrase is "patients" so analyzer assigns its semantic type, i.e. , *Patient or Disabled Group*, to the whole phrase.

Rule 3 ***If a phrase is "modifier (from the list of special modifiers) + [Anatomy] or [Physiology]" it is a phenotype name.***

We did not implement this rule directly. Instead we tried to teach it to our machine learning method. To help our machine learning method learn this rule, three binary features were added to the system: *Special Modifier*, *Anatomy* and *Physiology* to indicate whether a noun phrase is in the list of special modifiers [14] (see Appendix B) or in the semantic groups (SG) Anatomy or Physiology. Furthermore, some sentences containing this class of phenotypes (*Special Modifier* + [Anatomy] Or [Physiology]) were added to our training set.

Rule 4 ***If the singular form of a phrase is a phenotype the plural form is a phenotype, too.***

When the Result Merger searches for HPO phenotypes in the text, it searches for their singular and plural forms too and if it finds any of them, it will list the found term as a phenotype.

Rule 5 ***If the head of a phenotype candidate phrase is a phenotype, the whole phrase is a phenotype.***

This rule is considered in Result Merger: if the head of a noun phrase is found in HPO the whole noun phrase is tagged as a phenotype.

8.0.2 Adding features

To implement the rule-based system completely, Rule 3 was added as three features to the CRF, in addition to the features already possessed by BANNER. Finally, some other features which seemed to be helpful were added to the system. These features were tested several times and finally the best set of features were selected. These features include:

```
Phrase: "of Diamond-Blackfan anemia patients,"  ⟺  [Patient or Disabled Group]
>>>>> Syntax
msu
  prep([lexmatch([of]),inputmatch([of]),tag(prep),tokens([of])])
  mod([lexmatch([Diamond-Blackfan anemia]),inputmatch([Diamond,-,Blackfan,anemia]),tag(noun),
       tokens([diamond,blackfan,anemia])])
  head([lexmatch([patients]),inputmatch([patients]),tag(noun),tokens([patients])])
  punc([inputmatch([,]),tokens([])])
<<<<< Syntax
>>>>> Phrase
diamond blackfan anemia patients
<<<<< Phrase
>>>>> Candidates
Meta Candidates (8):
   812 Patients [Patient or Disabled Group]
   756 Anemia, Diamond-Blackfan [Congenital Abnormality]
   756 Diamond-Blackfan anemia (RPS19 gene) [Gene or Genome]
   645 ANAEMIA (Anemia) [Disease or Syndrome]
   645 Diamond [Element, Ion, or Isotope]
   645 Anemia (Genus Anemia) [Plant]
   645 Diamond (Diamond SPL Shape) [Qualitative Concept]
   574 anaemic [Finding]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (916):
   756 Anemia, Diamond-Blackfan [Congenital Abnormality]
   812 Patients [Patient or Disabled Group]
<<<<< Mappings
```

Figure 8.2: Analyzer Example

- ***Phenotype***: In Chapter 7 (the rule-based method) we discussed two different categories

of semantic types in the SG *Disorders*. ***Phenotype*** is a binary feature that indicates whether a noun phrase is in the *Phenotypes* category.

- ***Phenotype Candidates***: This feature is a binary feature that indicates whether a noun phrase is in the *Phenotype Candidates* category of SG *Disorders*.
- ***Special Modifier***: A binary feature that indicates whether a word is in the list of special modifiers [14].
- ***Anatomy***: A binary feature which means a noun phrase is in SG *Anatomy* or not.
- ***Physiology***: A binary feature which means a noun phrase is in SG *Physiology* or not.
- ***List Separator***: We found from many examples in the biomedical literature when a number of elements in a list are phenotypes, there is a good chance that the other elements are phenotypes too. The List Separator feature was added to designate the availability of list indicators (*and* or *comma*) in the sentence.
- ***Semantic type***: The semantic type which is assigned by MetaMap to noun phrases. This feature is null for other phrases.
- ***NP***: A binary feature which indicates whether a token is a part of a noun phrase.
- ***POS***: The part of speech of a token. This feature is a part of the BANNER feature set.
- ***Lemma***: The lemma of each token. This is available in BANNER.
- ***NPstart***: A binary feature to indicate whether a token is the first token in a noun phrase.
- ***NPend***: A binary feature to designate whether a token is the last token in a noun phrase.
- ***2, 3 and 4-character prefixes and suffixes***: This is a part of BANNER.
- ***2 and 3 character n-grams***: This is a part of BANNER.

Also, it should be remembered that BANNER makes a window of size two for each token, i.e. features of each token contain features of the two tokens before and the 2 tokens after it. We used the default configuration of BANNER. The NP feature comes from the MetaMap results and the POS feature is provided by BANNER. They do not align perfectly, but it is the task of the CRF to find a solution using these conflicting features.

8.0.3 Corpus

The most challenging problem for us in applying the machine learning method was the lack of an annotated specialized corpus of sufficient size. As no one before has used machine learning on phenotype name recognition, no corpus was available for us to train and test our system with. Therefore we had to make our own corpus with a sufficient number of sentences. However, making a large corpus is a very difficult and time consuming task. Therefore, we proposed an automatic method to annotate a corpus and then using semi-supervised learning we expanded the corpus.

Collecting the papers

To find the papers which are relevant to phenotypes, we started with two available databases: PubMed (2009) and BioMedCentral (2004). All HPO phenotypes were searched for in these databases and every paper which contained at least three different phenotypes was added to our collection. In this way we found 100 papers which were used to train the system. We had another 13 papers which had been collected for the development of the rule-based method (see Chapter 7) and were annotated manually. These 13 papers were used to test the system.

Annotating the corpus

As it was not possible for us to annotate the 100 papers manually, we started with the information provided by HPO to annotate our corpus and then using a self-training method the annotation process continued. This process is illustrated in Figure 8.3. First, HPO phenotypes were searched for in the set of papers and were tagged as phenotypes. These papers along with

their tags made our initial training corpus. When annotating the corpus, it should be noted that the last phase of the system (the Result Merger) was omitted because all HPO-annotated phenotypes were already annotated. The trained model was used to annotate the training set

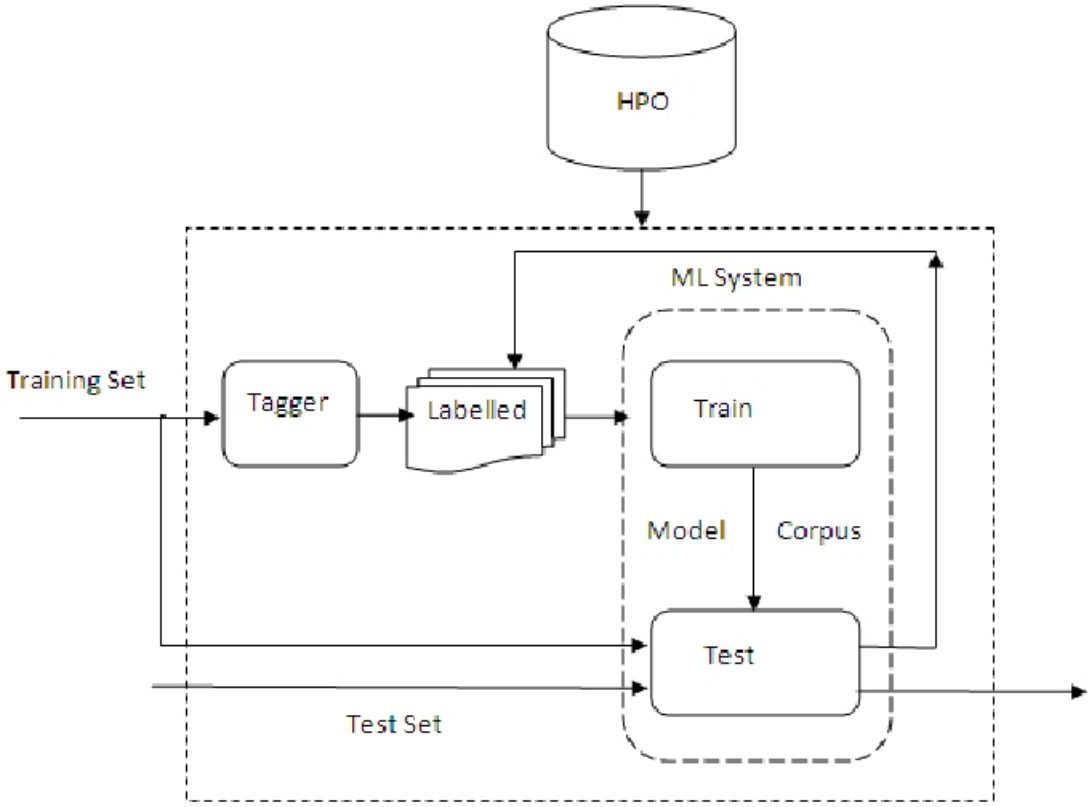

Figure 8.3: Process of corpus annotation

again. The newly found phenotypes were analyzed manually and the correct ones were added as annotations to the training set. Also, on each iteration, the system was tested using the test set to find out how many iterations would be sufficient for training the system. This process was repeated 3 times until we reached the results that we were satisfied with when testing the last model on the test set and when more iterations reduced the results. Doing more iterations could cause the overfitting problem.

One important point to mention is that we only included positive sentences (sentences with at least one phenotype) in our training set, because the number of negative sentences was far greater than positive sentences and it prevented the system from training efficiently. There-

fore, whenever new phenotype names were found, the number of sentences in the training set increased for the next iteration.

8.1 Evaluation

We compared the performance of our system against the rule-based system [55] explained in Chapter 7, which is the only specialized system for phenotype name recognition that we are aware of. The final training corpus which is made from 100 papers contains 2755 sentences and 4233 annotated phenotypes. All sentences in this corpus include at least one phenotype. A test set is collected from 13 papers and annotated manually includes 216 sentences and 373 phenotypes.

To evaluate the system, 10-fold cross validation has been used. Furthermore, the separate test set has been used to test the system. Table 8.1 gives the details of the results. The base system is our machine learning method ignoring the *Result Merger* module. Result Merger finds HPO phenotypes in text and adds them to the list of phenotypes. The results after using Result Merger are mentioned in the column labelled *HPO added*. The rule-based system has been tested using the test set and the results are displayed in Table 8.2. To calculate these results, a returned phrase is considered to be a true phenotype if its head contains a phenotype. For example, in the phrase "acute myloid leukemia" the head is "leukemia", a phenotype that is confirmed by its inclusion in HPO. However, in the phrase "Diamond-Blackfan anemia patients" the correct phenotype is: "Diamond-Blackfan anemia". If the system returns "Diamond-Blackfan anemia patients" as the phenotype, it is deemed false.

As this table demonstrates, the results are comparable with other named entity recognition systems which are specialized for finding other biomedical entities even though they may have larger training corpora. For example, BANNER has been trained and tested for finding gene names, using BioCreative 2 GM corpus containing 15,000 sentences (7500 for training and 7500 for testing) which is much larger than our current corpus. However our results are significantly better than BANNER's (Precision 85.09, Recall 79.06 and F-measure 81.96)[62].

Although our task is different, these results mean that our system is performing well.

In addition the machine learning method outperformed the rule-based method, even though the corpus is not fully annotated. We believe that if we had a fully annotated corpus the system would achieve an even better performance.

	Base system			HPO Added		
	Precision	Recall	F-measure	Precision	Recall	F-measure
10-Fold	82.83	68.13	74.35	86.89	98.33	92.25
Test Set	93.44	57.37	71.09	95.76	90.88	93.25

Table 8.1: System Evaluation Results

	Precision	Recall	F-measure
Machine Learning Method	95.76	90.88	93.25
Rule-Based Method	88.34	73.19	80.05

Table 8.2: Comparing the system with the Rule-Based Method

To have an idea of how well our annotation process works, we selected 100 random sentences from our corpus. Then, we tagged these sentences manually. There were 157 phenotypes in these 100 sentences. Our self-training method found 142 phenotypes and one of its phenotypes was incorrect, i.e. it missed 16 phenotypes. So, the annotation process misses about 10 percent of the phenotypes.

8.2 Discussion

Finding the best set of features is one of the most important parts of developing the system. Tables 8.3 and 8.4 show the role and importance of each feature. To illustrate the contribution of each feature in Table 8.3 we considered a small set of features as the basic set of features for our system. These features came from the rule-based system and are very important in signifying phenotype names in the text. Then, in each line a feature is added to the basic feature set until

we have the complete feature set in the last line. Adding some features (*Phenotype candidates*, *List separators*, *Semantic Types*, and *Lemma*) drops the results slightly but the results of the last feature set is better than the previous feature sets.

Analysing Table 8.3, it may seem that including some features is not necessary. Table 8.4 illustrates the role of each feature in a different way. In each line of Table 8.4 only one feature is ignored from the feature set and the system is tested using the separate test set. Note that the results are calculated without adding the HPO. As one can see, removing each feature reduces the results slightly. The exception to this modest reduction in performance is the removal of the *NP* feature which causes a significant drop in precision and recall, because not having NP information causes errors in finding *NP end* and *NP start*. In some cases (*Lemma*, *Phenotype candidates*, *NP start* and *NP end*, and *Physiology*) ignoring the feature causes small improvement in precision or recall. However the F-score is always less than the F-score of final results.

Features	Precision	Recall	F-Score
Phenotype, Anatomy, Physiology, Special Modifier	89.89	47.72	62.34
+Phenotype candidates	90.72	47.18	62.07
+List Separator	90.41	40.48	55.92
+Semantic type	90.24	49.59	64
+NP	90.24	49.59	64
+POS	91.15	55.22	68.77
+Lemma	92.72	54.69	68.79
+NP start, NP end	93.44	57.37	71.09

Table 8.3: Contribution of each additional feature

Reviewing the results, it has been found that both the rule-based and machine learning methods are dependent on MetaMap. MetaMap does make mistakes. MetaMap makes some errors in finding the boundaries of NPs and in determining the semantic types. NP boundaries and semantic types are features used by both methodologies, so the errors made by MetaMap

Ignored Feature	Precision	Recall	F-Score
Anatomy	92.82	55.49	69.45
Lemma	93.57	54.69	69.03
List Separator	91.66	56.03	69.54
Phenotype	92.54	56.56	70.20
NP	81.81	38.6	52.45
Phenotype Candidate	92.64	57.37	70.85
NP start and NP end	93.18	54.95	69.13
Physiology	93.21	55.22	69.35
POS	92.05	52.81	67.11
Semantic Type	91.89	54.69	68.56
Special Modifier	92.44	55.76	69.56

Table 8.4: Contribution of each feature

have effects on the performance of each system. However the rule-based system is more dependent on MetaMap output and errors in MetaMap output changes the results completely. But the machine learning system is more robust and it sometimes finds the correct phenotype names despite MetaMap errors. For example, consider the sentence *"Diamond-Blackfan anemia is a rare inherited bone marrow failure syndrome."*. The phrase *Diamond-Blackfan anemia* is a phenotype but MetaMap assigns the [Gene or Genome] semantic type to it, which is not in the SG *Disorders*. So the rule-based system does not tag it as a phenotype. The phrase *missing vertebrae* is another example of MetaMap errors. MetaMap does not consider this phenotype name as one NP. It separates this phenotype name into two phrases *missing* and *vertebrae*.

In addition, determining the boundary of an NP is very important in the rule-based system. MetaMap has an option to make larger NPs by merging simpler NPs. If we only use simple NPs, we cannot get larger phenotype names as one NP and the system will miss them. The phrase *Partial hypoplasia of the corpus callosum* is an example of a phenotype name with composite NPs. On the other hand if we use composite NPs, the head of the NP may change

and the semantic type of the NP may change as a result. This is problematic for the rule-based system. The phrase *the associations of facial dysmorphism* is an example. The word "*associations*" is the head of this phrase, so the semantic type [Mental Process] is assigned to it and it is not tagged as a phenotype name by the rule-based system.

On the other hand, there are some cases in which MetaMap assigns the correct semantic type to a phrase and found a good boundary for a phenotype name but the machine learning method does not mark it as a phenotype. For example "*arhinia*" is not tagged as a phenotype by the machine learning method in the following sentence *"These phenotypes may resemble that of the only confirmed case of an individual with a lethal compound heterozygous PAX6 mutation and may include anophthalmia, arhinia and severe central nervous system defects"* although MetaMap assigns the semantic type [Congenital Abnormality] (which is in the category of Phenotypes) to it.

Table 8.5 shows how many false negatives and true positives are available in the final results for both the machine learning and the rule-based methods. Table 8.6 illustrates the percent of these errors caused by MetaMap or the boundary of noun phrases.

	Rule-based	Machine Learning
True positives	273	339
False negatives	100	34

Table 8.5: Number of TPs and FNs in each method

	Rule-based	Machine Learning
MetaMap errors	37%	11.76%
NP boundary errors	26%	8.82%

Table 8.6: Analysis of NPs

Comparing the precision errors gave interesting observations. There are no common errors between the two systems. False positive errors in the rule-based system were caused by the rules not being discriminating enough. For each returned phrase, one of the rules produced

that phrase. But there are exceptions to each rule that can cause these false positive errors. The exceptions to the rules cause fewer problems for the machine learning system. The machine learning system was able to learn all of these exceptions. For the false phenotypes returned by the machine learning system, analysis indicates that none of these would be suggested by application of a rule in the rule-based system, explaining why there are no common errors.

Chapter 9

Improving Phenotype Name Recognition

The proposed phenotype Name Recognition systems are dependent on MetaMap. MetaMap output provides the basic information for extracting the noun phrases (NP), their semantic types (ST) and finally deciding whether they are phenotypes or not. One of the important rules in the proposed (rule-based and machine learning) systems declares how to map an ST to an NP. Sometimes MetaMap cannot find the whole NP in UMLS so it breaks the NP into pieces and maps one ST to each part. According to the second rule in the rule-based system:

Rule 2 The semantic type of a noun phrase is the semantic type assigned by MetaMap to its head.

We call this rule the ***Head rule***. Figure 9.1 shows an example of such an NP. The phrase *"an autosomal disorder"* is broken into two parts; *"autosomal"* and *"disorder"* and each part has a different ST. According to MetaMap output *"disorder"* is the head of this phrase, so the ST of the whole phrase would be *"Disease or Syndrome"*.

However sometimes MetaMap makes errors or finding STs based on the ***Head rule*** does not work. Figure 9.2 shows an example of an NP which includes a phenotype. Consider the sentence *"In 40 to 50% of Diamond-Blackfan anemia patients, congenital abnormalities mostly in the cephalic area and in thumbs and upper limbs have been described."*. Diamond-Blackfan anemia is a phenotype but according to MetaMap output *"Diamond-Blackfan anemia patients"* is an NP which is completely correct. However the head of this NP is "*patients*" and

```
Phrase: "an autosomal disorder"
>>>>> Syntax
msu
 det([lexmatch([an]),inputmatch([an]),tag(det),tokens([an])])
 mod([lexmatch([autosomal]),inputmatch([autosomal]),tag(adj),
    tokens([autosomal])])
 head([lexmatch([disorder]),inputmatch([disorder]),tag(noun),
 tokens([disorder])])
<<<<< Syntax
>>>>> Phrase
autosomal disorder
<<<<< Phrase
>>>>> Candidates
Meta Candidates (Total=2; Excluded=0; Pruned=0; Remaining=2)
  861   Disorder (Disease) [Disease or Syndrome]
  694   autosomal (Autosome) [Cell Component]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (888):
  694   autosomal (Autosome) [Cell Component]
  861   Disorder (Disease) [Disease or Syndrome]
<<<<< Mappings
```

Figure 9.1: MetaMap output for *"an autosomal disorder"*

based on the ***Head rule*** the ST of this phrase is *"Patient or Disabled Group"* and the system is not able to recognize *"Diamond-Blackfan anemia"* as a phenotype.

As we mentioned in the previous Chapter, about 8.82% of the errors in our machine learning based system are caused by the incorrect determination of NP boundaries. Figure 9.3 illustrates an example of an error made by MetaMap which results in detecting a wrong NP boundary. The sentence, *"One of them (GUE) presented learning disabilities while this information was unavailable for N2603 and OLI who were 4 and 3 years old at the examination time, respectively."* is processed by MetaMap. As you can see in Figure 9.3, MetaMap has decided that *"presented learning disabilities"* is an NP which is clearly wrong. This error causes problems in finding the phenotypes. Based on this output, *"presented learning disabilities"* is a phenotype while we know that the correct phenotype is *"learning disabilities"* and *"presented"* is the verb.

In this Chapter we propose solutions for these kinds of problems which directly affect the performance of the phenotype name recognition systems.

```
Phrase: "of Diamond-Blackfan anemia patients,"
>>>>> Syntax
msu
 prep([lexmatch([of]),inputmatch([of]),tag(prep),tokens([of])])
 mod([lexmatch([Diamond-Blackfan anemia]),inputmatch([Diamond,-,Blackfan,anemia])
     ,tag(noun),tokens([diamond,blackfan,anemia])])
 head([lexmatch([patients]),inputmatch([patients]),tag(noun),tokens([patients])])
 punc([inputmatch([,]),tokens([])])
<<<<< Syntax
>>>>> Phrase
diamond blackfan anemia patients
<<<<< Phrase
>>>>> Candidates
Meta Candidates (Total=9; Excluded=1; Pruned=0; Remaining=8)
  812   Patients [Patient or Disabled Group]
  756   Diamond-Blackfan anemia (Congenital pure red cell aplasia)
        [Congenital Abnormality]
  756   Anemia, Diamond-Blackfan [Congenital Abnormality,Disease or Syndrome]
  756   Diamond-Blackfan anemia (RPS19 gene) [Gene or Genome]
  645   Anaemia (Anemia) [Disease or Syndrome]
  645   Diamond [Element, Ion, or Isotope]
  645   Anemia (Genus Anemia) [Plant]
  645   Diamond (Diamond SPL Shape) [Qualitative Concept]
  574 E anaemic [Finding]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (916):
  756   Anemia, Diamond-Blackfan [Congenital Abnormality,Disease or Syndrome]
  812   Patients [Patient or Disabled Group]
<<<<< Mappings
```

Figure 9.2: MetaMap output for *"Diamond-Blackfan anemia patients"* in the sentence *"In 40 to 50% of Diamond-Blackfan anemia patients, congenital abnormalities mostly in the cephalic area and in thumbs and upper limbs have been described."*

9.1 Proposed method

As we mentioned earlier we want to consider two types of problems and propose solutions for each. The problems are caused when:

1. A phenotype name modifies the head of the NP containing it.

2. MetaMap identifies a wrong boundary for the NP containing a phenotype name.

```
Phrase: "presented learning disabilities"
>>>>> Syntax
msu
 mod([lexmatch([presented]),inputmatch([presented]),tag(adj),tokens([presented])])
 head([lexmatch([learning disabilities]),inputmatch([learning,disabilities]),
    tag(noun), tokens([learning,disabilities])])
<<<<< Syntax
>>>>> Phrase
presented learning disabilities
<<<<< Phrase
>>>>> Candidates
Meta Candidates (Total=9; Excluded=4; Pruned=0; Remaining=5)
  901   Learning Disabilities [Mental or Behavioral Dysfunction]
  867 E Learning disability (Learning disability-specialty)
                    [Biomedical Occupation or Discipline]
  827   learning (Knowledge acquisition) [Educational Activity]
  827   Learning [Mental Process]
  827   Disabilities (Disability) [Finding]
  743 E Disabled (Disabled Persons) [Patient or Disabled Group]
  743 E Disabled [Qualitative Concept]
  660   Presented (Presentation) [Idea or Concept]
  627 E Present [Quantitative Concept]
<<<<< Candidates
>>>>> Mappings
Meta Mapping (901):
  660   Presented (Presentation) [Idea or Concept]
  901   Learning Disabilities [Mental or Behavioral Dysfunction]
<<<<< Mappings
```

Figure 9.3: MetaMap output for *"learning disabilities"* in the sentence *"One of them (GUE) presented learning disabilities while this information was unavailable for N2603 and OLI who were 4 and 3 years old at the examination time, respectively."*

9.1.1 Empty heads

The first problem occurs when the semantic type of the head is assigned to the whole NP and the system cannot recognize the phenotypes embedded inside the NP. To solve this problem we used the concept of ***empty heads***.

Empty heads [41] is a widespread phenomenon in biomedical text. When an NP has an empty head, its head semantic type does not represent the semantic type of some of the embedded NPs. In our system, we refer to a head as an Empty head, when the head of a NP prevents the system from recognizing the phenotype name embedded inside it. Figure 9.2 shows an example of such a phenomenon, where *patients* is the head. But if we assign its semantic type to the whole NP, the system cannot detect the phenotype name inside the NP.

As our purpose is to extract phenotype names we only consider the empty heads problematic for detecting phenotypes. We found the following terms in our corpus which cause the empty head problem (In a larger corpus there might be other empty heads.).

1. Affected (patients, individuals, women, children, males, etc.) e.g. *Diamond-Blackfan anemia-affected patients*

2. Patients e.g. *Amyotrophic lateral sclerosis patients*

3. Cells e.g. *Cancer cells*

4. Spectrum e.g. *Holoprosencephaly spectrum*

5. Cases e.g. *Diamond Blackfan anemia cases*

6. Type (enumeration) e.g. *Neurofibromatosis type 1*

These empty heads can be classified into two categories:

- In the first five examples the head of the NP is not a part of the phenotype.

- In the last example the head is a part of the phenotype.

To solve the problem related to the first class of empty heads, whenever one of these terms is the head of an NP, we just map its semantic type to the head and assign the semantic type of the previous term to the remainder of the NP. To solve the problem related to the word *"type"*, whenever it is the head of an NP, its semantic type is ignored and the semantic type of the previous term is assigned to the whole NP.

Table 9.1 represents examples in which the mapped STs improved by ignoring the empty heads. These examples are phenotypes but their semantic types before ignoring the empty heads is not in SG *Disorders* so the system will not recognize them as phenotypes. However, after applying the solutions their STs fall into SG *Disorders*.

NP	ST before ignoring empty heads	ST after ignoring empty heads
Wolf-Hirschhorn syndrome patients	Patient or Disabled Group	Disease or Syndrome
X-linked ocular albinism type 1	Classification	Disease or Syndrome
the Holoprosencephaly spectrum	Conceptual Entity	Congenital Abnormality

Table 9.1: Examples showing the effectiveness of ignoring empty heads.

9.1.2 NP boundary

A problem occurs when MetaMap does not recognize the correct boundary of an NP. This problem has two consequences:

- A large NP is broken into several smaller NPs (for example *"Diamond-Blackfan anemia"* is broken into *"Diamond"* and *"Blackfan anemia"*).
- An example of this problem is shown in Figure 9.3 where a term ("*presented*") which does not belong to an NP is considered as a part of it.

We considered three solutions for this problem:

1. Changing the MetaMap parser: MetaMap uses a shallow parser called *Specialist minimal commitment*. We thought about changing its source code to use another parser which makes fewer mistakes. MetaMap source code is in SICStus Prolog which is a commercial version of Prolog. So this solution would have incurred an expense that was not in budget and would have also been time-consuming to understand the source code.
2. Extracting NPs using another parser and giving only the extracted NPs (without their sentences) to MetaMap: This solution ignores the dependency of the MetaMap word sense disambiguation module on other parts of a sentence to extract the correct sense and the semantic type of the NP.

3. Extracting the boundary of NPs using another parser and giving the whole sentence with marked-up NPs to MetaMap.

The third solution seemed reasonable but the question was how to mark up the NPs in the MetaMap input sentences: One idea we considered was to bracket the NPs with parentheses. Table 9.2 shows a part of a test we performed to understand if bracketing the boundary of NPs in MetaMap input would help MetaMap make better decisions in detecting appropriate NPs. As this table illustrates, when the original sentences are processed by MetaMap the detected NPs are not what we expected and it makes phenotype name recognition difficult. However, when the NPs are wrapped in brackets, MetaMap extracts the correct NPs. As a result MetaMap maps better STs to the NPs and it makes phenotype name recognition more effective.

9.2 Implementation

Implementing the idea of ignoring empty heads was straightforward. MetaMap provides the syntactic information of each sentence made by its parser. Using this information we can check whether the head is in our list of empty heads and, if so, do the appropriate action based on which class of empty heads it belongs to.

To implement the bracketing idea we added a pre-processing step to our system to bracket the text before giving it as an input to MetaMap. Two available NLP tools have been used. Figure 9.4 illustrates a partial block diagram of our system after adding these tools for the pre-processing step. *PostMed* extracts and annotates sentences from the text. Then BLLIP parses each sentence individually and provides the syntactic structure of each sentence. Finally *Extract NPs* which is implemented by the author, processes the BLLIP output and brackets the NPs. The NPs tagged by BLLIP are extracted and bracketed by *Extract NPs* in two ways:

1. Complete bracketing: If an NP is inside another NP, both of them are bracketed. In case only one token is inside an NP it is not bracketed.

2. Biggest NP: In the case of embedded NPs, only the outermost one is bracketed.

Bracketed NP	MetaMap returned NP before bracketing	MetaMap returned NP after bracketing
Sentence: *Patients with mild to moderate bone deformities and variable short stature are classified as Osteogenesis imperfecta type IV.*		
(Osteogenesis imperfecta type IV)	*classified as Osteogenesis imperfecta type IV*	*Osteogenesis imperfecta type IV*
Sentence: *Clinical signs include a typical facial appearance, resembling the "Greek warrior helmet" profile, mental retardation, severe growth delay, hypotonia, congenital heart malformations, midline defects, such as cleft palate and hypospadias, ocular colobomas, renal abnormalities and seizures.*		
(severe growth delay)	Two NPs: *severe*, *growth delay*	*severe growth delay*
Sentence: *A sister (II-2, bilateral cleft lip/palate, microcephaly), brother (II-3, cervical rachischisis, missing vertebrae) showed characteristics within the Holoprosencephaly spectrum.*		
(bilateral cleft lip/palate)	Two NPs: *bilateral cleft lip*, */palate*	*bilateral cleft lip/palate*
(missing vertebrae)	Two NPs: *missing*, *vertebrae*	*missing vertebrae*
Sentence: *Diamond-Blackfan anemia is a rare inherited bone marrow failure syndrome (five to seven cases per million live births) characterized by an aregenerative, usually macrocytic anemia with an absence or less than 5% of erythroid precursors (erythroblastopenia) in an otherwise normal bone marrow.*		
(an aregenerative, usually macrocytic anemia)	*characterized by an aregenerative, usually macrocytic anemia*	*an aregenerative, usually macrocytic anemia*

Table 9.2: Experimental results showing the effectiveness of bracketing the MetaMap input to affect its NP boundary detection.

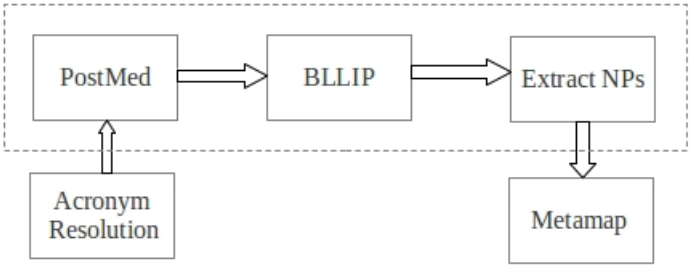

Figure 9.4: Block diagram of the pre-processing step.

The following presents an example of bracketing performed by *Extract NPs*:

- The original sentence: *Patients with mild to moderate bone deformities and variable short stature are classified as Osteogenesis imperfecta type IV.*
- BLLIP result: (S1 (S (S (NP (NP (NNS Patients)) (PP (IN with) (NP (NP (ADJP (JJ mild) (TO to) (JJ moderate)) (NN bone) (NNS deformities)) (CC and) (NP (JJ variable) (JJ short) (NN stature))))) (VP (VBP are) (VP (VBN classified) (PP (IN as) (NP (NN Osteogenesis) (NN imperfecta) (NN type) (CD IV)))))) (. .)))
- Complete bracketing: *(Patients with ((mild to moderate bone deformities) and (variable short stature))) are classified as (Osteogenesis imperfecta type IV).*
- Biggest NP: *(Patients with mild to moderate bone deformities and variable short stature) are classified as (Osteogenesis imperfecta type IV).*

9.3 Results and discussion

To evaluate the performance of the proposed pre-processing step, the same training and test sets discussed in Chapter 8 have been used. The training set is made from 100 papers and contains 2755 sentences. The test set is collected from 13 papers and includes 216 sentences.

There are two approaches in evaluating named entity recognition systems: loose evaluation and strict evaluation [35]. Under a strict evaluation scheme, a method needs to find the exact

term specified by the annotator, with exact left and right boundaries. Under loose evaluation, the found named entity only needs to overlap with the specified term.

The named entity boundary detection task is one of the most difficult tasks in NER [65]. To interpret the results correctly, it is worth noting that in strict evaluation, performance drops significantly because the system is punished twice for every prediction with a different boundary, once because the predicted named entity is not true (false positive) and once because the correct term is not found (false negative). GAPSCORE [16] (a method for finding gene and protein names in text) encountered a 24.9 percentage points difference in F-measure between loose and strict evaluation results. Hakenberg et al. [42] proposed a machine learning system for extracting gene and protein names from literature. They found a 10 percentage point difference in their strict and loose evaluation results.

Chapters 7 and 8 only report the loose evaluation results of our proposed machine learning method for extracting phenotype names; however, in this Chapter we are trying to improve the strict evaluation results as well as the loose evaluation results. Table 9.3 illustrates the loose and strict evaluation results of our previuosly proposed phenotype name recognition system before adding HPO to it. As we expected, the strict evaluation performance is less than the loose evaluation one.

	Precision	Recall	F-measure
Loose evaluation using MetaMap 2012	93.33	57.37	71.09
Strict evaluation using MetaMap 2012	86.99	50.78	64.12
Loose evaluation using MetaMap 2013	93.64	57.85	71.51
Strict evaluation using MetaMap 2013	87.28	53.92	66.66

Table 9.3: Strict and loose evaluation results for our base phenotype name recognition system.

The reported results in Chapter 8 were achieved using MetaMap 2012. While we were working on improving the performance of our system, MetaMap 2013 was released. MetaMap 2013 solved some of our problems and some other problems remained unsolved. For example in Table 9.2 problems with *"severe growth delay"* and *"bilateral cleft lip/palate"* are solved but

	Without HPO			HPO Added		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Base system	93.64	57.85	71.51	95.85	90.83	93.27
Empty head	94.33	52.21	67.21	96.54	87.17	91.61
Complete bracketing	93.30	54.71	68.97	95.77	89	92.26
Biggest NP	94.64	55.49	69.96	96.63	90.31	93.36
Empty head+ Complete bracketing	93.39	55.49	69.61	95.78	89.26	92.40
Empty head+ Biggest NP	94.06	53.92	68.54	96.34	89.79	92.94

Table 9.4: The contribution of each solution in the loose evaluation results to the results provided by the base system using MetaMap 2013.

	Without HPO			HPO Added		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Base system	87.28	53.92	66.66	91.95	89.79	90.85
Empty head	88.31	49.47	63.42	93.05	87.69	90.29
Complete bracketing	87.94	51.57	65.01	92.58	88.21	90.34
Biggest NP	89.28	52.35	66.00	93.46	89.79	91.58
Empty head+ Complete bracketing	87.22	51.83	65.02	92.09	88.48	90.24
Empty head+ Biggest NP	88.73	51.57	65.23	93.24	90.31	91.75

Table 9.5: The contribution of each solution in the strict evaluation results to the results provided by the base system using MetaMap 2013.

the other problems remain. Therefore the performance of our phenotype recognition system improved somewhat.

NP	ST before ignoring empty heads	ST after ignoring empty heads
Diamond Blackfan anemia	Congenital Abnormality, Disease or Syndrome	Congenital Abnormality, Disease or Syndrome
Bilateral cleft lip/palate	Disease or Syndrome	Disease or Syndrome
Diamond Blackfan anemia in *"Diamond Blackfan anemia patients"*	Patient or Disabled Group	Congenital Abnormality, Disease or Syndrome
Neurofibromatosis type 1 in *"our group of Neurofibromatosis type 1 patients"*	Patient or Disabled Group	Neoplastic Process

Table 9.6: Examples of the phenotypes found in the base system but after ignoring the empty heads, system could not extract them.

Tables 9.4 and 9.5 represent the contribution of our proposed solutions on loose and strict evaluation results of the phenotype recognition system. In both cases, ignoring empty heads improves the precision somewhat but drops the recall. Although it improves the correctness of STs assigned to NPs with empty heads, it is not effective. Our test results show that after ignoring the empty heads the number of found phenotypes decreased. Furthermore no phenotype was found which was not available in the list of phenotypes extracted by the base system. Table 9.6 presents some examples of the phenotypes found in the base system but after ignoring the empty heads, the system could not extract them. As seen in the table, in all cases the STs stayed the same or improved after ignoring the empty heads. The reason might be because the frequency of NPs containing empty heads in our training set is less than 1%, and these are the only cases of phenotypes where NPs heads are not a part of the phenotype. We are trying to teach the system an exception without having enough related training examples. Apparently, it just confuses the system and decreases the performance.

Between the two bracketing solutions, Complete bracketing and Biggest NP, the perfor-

mance of Biggest NP is better. The best result of applying Biggest NP is achieved in strict evaluation where precision is improved 2% and recall dropped only 1.57% before adding HPO and after adding HPO phenotypes to the list of found phenotypes precision improved 1.51% and recall stays the same. This result was expected because:

- As shown in the Table 9.2 the remaining problems in MetaMap 2013 are mostly related with its confusion between verbs and past participles or gerunds. Bracketing the biggest NPs helps to solve this problem. Based on the results we can conclude that in the case of embedded NPs, MetaMap determines better NP boundaries without any bracketing. Actually, in some cases the bracketing of inner NPs is more confusing. For example, consider the sentence *"De novo mutation p.V137RfsX18 was detected in a 10-month-old girl with semilobar Holoprosencephaly (partial hypoplasia of the corpus callosum)."*

  - When biggest NPs are bracketed: *(De novo mutation p.V137RfsX18) was detected in (a 10-month-old girl with semilobar Holoprosencephaly (partial hypoplasia of the corpus callosum))*.
  - After complete bracketing: *((De novo mutation p.V137RfsX18) was detected in ((a 10-month-old girl) with ((semilobar Holoprosencephaly) (((partial hypoplasia) of (the corpus callosum))))).*
  - MetaMap 2013 considers *"partial hypoplasia of the corpus callosum"* as one NP after processing the non bracketed sentence.
  - MetaMap 2013 considers *"partial hypoplasia of the corpus callosum"* as one NP after processing the biggest NPs bracketed sentence.
  - MetaMap 2013 considers *"partial hypoplasia of the corpus callosum"* as two separate NPs (*partial hypoplasia* and *corpus callosum*) after processing the completely bracketed sentence.

- Strict evaluation is more sensitive to the NP boundaries. Solving the NP boundary problem helps the system to recognize a better boundary for phenotypes.

Although ignoring empty heads did not improve the system performance, combining it with Biggest NP provides the best results after adding HPO. These two solutions together improve the precision somewhat but the recall reduces slightly but recovers when the information from HPO is added. Adding HPO causes a significant increase in both the precision and the recall. This is the only case that recall is better than the base system. It seems that the phenotypes were missed before adding the HPO are not significant because HPO already include them. In addition these two solutions find more phenotypes which are not available in HPO in comparison with the base system.

Chapter 10

Phenotype–Genotype Relation Extraction

In Chapters 6, 7, 8 and 9 methods for extracting genotype names and phenotype names from biomedical text have been discussed. This Chapter covers the last step in achieving the goal of this thesis: extracting relationships between phenotypes and genotypes.

Finding the relationships between entities (mostly proteins) from information contained in the biomedical literature has been studied extensively and many different methods to accomplish these tasks have been proposed. However, we are not aware of any other system dedicated to extracting phenotype-genotype relations, therefore no information about the nature of these relations and how they differ from relations between other biomedical entities was available to us. Furthermore, there was no annotated data to start with.

Using computational linguistics or rule-based approaches requires analyzing a large set of annotated data to capture linguistic and semantic rules that capture relationships between genotypes and phenotypes and often it demands prior knowledge about the domain. This process is very time consuming and the lack of sufficient quantity of labelled data made it impossible for us.

Applying machine learning based methods could alleviate one of the problems because it does not need extensive prior knowledge about biomedicine and it can prove to be faster because manual processing of the data is not required. However, supervised machine learning is impossible without having labelled data to be used for training a model. The lack of labelled

data still posed a significant problem for us. Therefore, we decided to take advantage of semi-supervised learning.

Semi-supervised learning can be done with a small labelled set but we did not have even a small set of labelled data. Instead we have proposed a semi-automatic method for creating a small set of labelled data by applying two available protein-protein interaction methods [34] to the phenotype-genotype relationship problem. Then using this seed in a self-training framework, a machine learning model has been trained. We will cover the details of each step in the remainder of this chapter.

10.1 Curating the data

As mentioned before we did not have access to any prepared data for the phenotype-genotype relation extraction task, so our first task was to collect enough data containing phenotype and genotype names. Three sources of data have been used in this project:

- We made a corpus for the phenotype name recognition task [56], which is designated as *MKH* here. This Corpus is comprised of 113 full papers and 2971 sentences. Genotype and phenotype names in this corpus were located using the proposed methods (Chapters 6 and 8).

- PubMed was queried for "Genotype and Phenotype and correlation" and 5160 paper abstracts were collected. Genotype and phenotype names were located using the previously explained methods (Chapters 6 and 8).

- Collier et al. [21] generated and made available to us the *Phenominer* corpus which contains 112 PubMed abstracts. The annotation was carried out with the same experienced biomedical annotator who accomplished the GENIA corpus [80] tagging. Both phenotypes and genotypes are annotated in this corpus. *Phenominer* contains 1976 sentences with 1611 genotypes and 472 phenotype candidates. However there are two issues with this corpus:

- The phenotypes at the cellular level are labelled in this corpus while according to the definition of phenotype in this thesis we do not consider them as phenotypes.
- Generic expressions (e.g. , gene, protein, expression) referring to a genotype or a phenotype earlier in the text are tagged in this corpus as phenotypes and genotypes. For example *locus* is tagged as a genotype in the following sentence: *"Our original association study focused on the role of IBD5 in CD; we next explored the potential contribution of this* ***locus*** *to UC susceptibility in 187 German trios."*

Only sentences with both phenotype and genotype names have been selected from the above resources to comprise our data and the remaining sentences have been ignored. In this way we have collected 460 sentences from the *MKH* corpus, 3590 sentences from the *PubMed* results and 207 sentences from *Phenominer*. These sentences made our initial set of sentences.

All the sentences are represented by the IOB label model (Inside, Outside, beginning). The selected sentences go through the following pre-processing step. The phenotype names and genotype names are tagged by their token offset from the beginning of each sentence. This step is necessary because sometimes an entity name repeats in a sentence and it gets confusing for both labelling the data and understanding the relationships. To understand this tagging step, consider the following sentence:

- Common esr1 gene alleles are unlikely to contribute to obesity in women, whereas a minor importance of esr2 on obesity cannot be excluded.

Assume that we labelled this sentence and a relation between *obesity* and *esr2* is annotated. The system gets confused about *obesity*. It is not clear which *obesity* has a relation with *esr2*, the first one or the second one. Although these words are the same, their grammatical roles and the relations with other tokens in the sentence are completely different. Therefore, this sentence is changed to the following form after pre-processing (note punctuation is considered a token.):

- Common esr1 gene alleles-4 are unlikely to contribute to obesity-10 in women, whereas a minor importance of esr2-19 on obesity-21 cannot be excluded.

10.1.1 Training set

For supervised learning the training set must consist of labelled data. Even for semi-supervised learning, the initial training set must consist of labelled data. However, at the beginning of the project we did not have any labelled data. Preparing labelled data would require hiring annotators knowledgeable in biomedicine. This proves to be expensive and very time-consuming. Neither of the resources, research money nor time, were in abundance. So, we decided instead to use two available protein-protein interaction extraction tools on our data and use their best outputs as our labelled training set.

Ibn Faiz [34] proposed a rule-based (see Section 2.3.2) and a machine learning-based system (see Section 2.3.3) for extracting relations between pairs of proteins. Although these tools were not implemented for extracting phenotype-genotype relations, they are able to detect the more general types of relationships which are similar between pairs of proteins and phenotype-genotype pairs. For example the rule-based system is able to find the following relationships:

- ENTITY1 Relation ENTITY2; e.g., GENOTYPE *causes* PHENOTYPE
- Relations in which the entities are connected by one or more prepositions:
  - ENTITY1 *Relation (of | by | to | on | for | in | through | with)* ENTITY2; e.g., PHENOTYPE *is associated with* GENOTYPE
  - *(PREP | REL | N)+(PREP)(REL | PREP | N)** ENTITY1 *(REL | N | PREP)+* ENTITY2; where PREP is any preposition, REL is any relation term, N is any noun. e.g., *Activation of* PHENOTYPE *by* GENOTYPE
  - *Relation (of | by | to | on | for | in | through | with | between)* ENTITY1 and ENTITY2, e.g., *Correlation between* GENOTYPE *and* PHENOTYPE.
- ENTITY1 (/ | \ | −) ENTITY2; e.g., GENOTYPE/PHENOTYPE *Correlation*.

These protein-protein interaction (PPI) extraction tools are available to us and we modified them as follows. The rule-based method has a list of relation terms demonstrating a relationship

between two proteins in a sentence. Rindflesch et al. [88] mentioned a list of 20 verbs and two prepositions (*in* and *for*) which encode a relation between a genetic phenomenon and a disorder (Appendix C). We added these words to the list of relation terms found in Ibn Faiz's original system.

Our initial corpus is pre-processed and then is separately processed by Ibn Faiz's rule-based and machine learning-based PPI extraction tools. Each of these tools finds some relations in the input sentences. After the results are compared those sentences that contain at least one agreed- upon relationship [1] are initially considered as the training set. From the original corpus, 519 sentences comprised the initial training set as the result of this process. However, as our utilized tools were originally developed for finding protein-protein interactions, not phenotype-genotype relationships we could not be certain that even their similar results produced correctly labelled examples. Therefore, the initial training set was further processed manually. Several interesting issues were observed.

1. Some sentences do not state any relationship between the annotated phenotypes and genotypes. Instead, these sentences only explain the aim of a research project. However, these sentences are labelled as containing a relationship by both tools. e.g. , *"The present study was undertaken to investigate whether rare variants of TNFAIP3 and TREX1 are also associated with systemic sclerosis."*

2. The negative relations stated with the word "no" are considered positive by both tools. e.g. , *"With the genotype/phenotype analysis , no correlation in patients with ulcerative colitis with the MDR1 gene was found."*

3. Some sentences from the *Phenominer* corpus are substantially different compared to other sentences, because of the two issues we discussed earlier about this corpus. The phenotypes below the cellular level have different relations with genotypes. For exam-

[1] Phenotype-genotype pairs that have a relationship are the positive instances. Phenotype-genotype pairs that do not have a relationship are the negative instances. The sentences mentioned have both positive and negative instances.

ple, they can change genotypes while the supercellular-level phenotypes are affected by genotypes and are not capable of causing any change to them.

4. Some cases have both tools making the same mistakes: suggesting incorrect relationships (i.e. , negative instances are suggested as positive instances) or missing relationships (i.e. positive instances are given as negative instances).

After making corrections (see issues 1 and 4) and deleting sentences with issues 1 and 3, 430 sentences remained in the training set. These corrections and deletions were made by the author. This data set is skewed: there are many fewer negative instances than positive instances. To address this imbalance, 40 sentences without any relationships have been selected manually and are added to the training set. Also, to increase the training set size 39 additional sentences have been labelled manually and have been added to the training set. The final training set has 509 sentences. There are 576 positive instances and 269 negative instances.

10.1.2 Test set

To select the sentences to be included in the test set, the results from processing our initial set with the two PPI tools have been used. In some cases both tools extract relations from the same sentence but the relations differ. For example in sentence *"Common esr1 gene alleles-4 are unlikely to contribute to obesity-10 in women, whereas a minor importance of esr2-19 on obesity-21 cannot be excluded."*, the machine learning-based tool finds a relationship between *esr2-19* and *obesity-21* but the rule-based tool claims that there is also a relationship between *esr1 gene alleles-4* and *obesity-10*. Since we were confident that this type of sentence would provide a rich set of positive and negative instance, this type of sentence is extracted to make our initial test set of 298 sentences.

In order for the test set to provide a reasonable evaluation of the trained model, the sentences must be correctly labelled. A biochemistry graduate student was hired to annotate the initial test set. Pairs of genotypes and phenotypes are extracted from each sentence and her task was to indicate whether there was any relationship between them. We also asked her to give

```
PS699 Alterations in tumour suppressor genes (p53-7, Rb-9) are associated instead with
the most aggressive and poorly differentiated forms of thyroid cancer-24 , indicating
that, in the thyroid tumourigenic process, they represent late genetic events .

!Y      //[PHENOTYPE-thyroid cancer-24] [GENOTYPE-Rb-9]
!Y      //[PHENOTYPE-thyroid cancer-24] [GENOTYPE-p53-7]

#Comments: Genotypes should be alterations in p53 or alterations in Rb
#Verb: are (associated)
```

Figure 10.1: An example of an annotated sentence.

any comments about the correctness of the extracted genotypes or phenotypes and indicate the verb that creates the relationship between each pair. Figure 10.1 presents a sample annotated sentence.

Issues 1 and 3 discussed in Section 10.1.1 have been observed by the annotator in some of the sentences. Also, there were some cases where she was not sure if there was a relationship or not. Furthermore, she disagreed with several annotated phenotypes and genotypes. After deleting some problematic sentences and fixing several problems the final test set comprising 244 sentences has been formed.

10.1.3 Unlabelled data

After choosing the training and testing sentences from the initial set of sentences the remaining sentences have been used as unlabelled data. The unlabelled set contains 3440 sentences.

10.2 Training a model

Now that we have a labelled training set, it is possible to train a machine learning model using a supervised method to be evaluated on the test set. On the other hand, because the training set is so small, using supervised learning does not lead to good results (precision: 76.47, recall: 77.61), so a self-training algorithm has been proposed to improve the results.

10.2.1 Machine learning method

We applied the machine learning method proposed by Ibn Faiz [34] for PPI extraction to our task. This method uses a maximum entropy classifier and converts the relation extraction problem to a binary classification problem. A genotype-phenotype pair is represented by a set of features derived from a sentence. The features are classified into three different groups: dependency features extracted from the dependency path between the two entities, syntactic features derived from the syntax tree of the sentence, and surface features obtained directly from the raw text. Tables 10.1 and 10.2 represent the list of features.

Figure 10.2 shows the dependency tree produced by the Stanford dependency parser[2] for the sentence *"The association of Genotype1 with Phenotype2 is confirmed."*. Using this dependency tree, the dependency path between the two entities, which contains important information, is extracted. The dependency path between the phenotype and the genotype in this figure is "Genotype1-*prep_of*-association-*prep_with*-Phenotype2". *Association* is the relation term in this path and *prep_of* and *prep_with* are the dependency relations related to it. The presence of a relation term can be a signal for the existence of a relationship and its grammatical role along with its relative position gives valuable information about the entities involved in the relationship. Sometimes two entities are surrounded by more than one relation term. The key term feature is introduced to find the relation term which best describes the interaction. Ibn Faiz [34] used the following steps to find the key term, when one step fails the process continues to the next step but if the key term is found in one step the following steps are ignored.

1. Any relation term that occurs between the entities and dominates them both in the dependency representation is considered as the key term.

2. A word is found that appears between the entities, has a child which is a relation term and dominates the two entities. That child is considered as the key term.

3. Any relation term that occurs on the left of the first entity or on the right of the second entity and dominates them both in the dependency representation is considered as the

[2]http://nlp.stanford.edu/software/stanford-dependencies.shtml

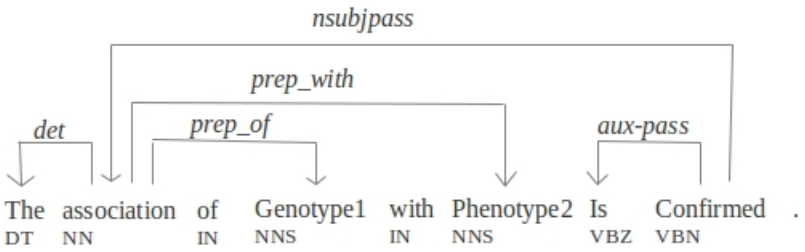

Figure 10.2: Dependency tree related to the sentence *"The association of Genotype1 with Phenotype2 is confirmed."*

key term.

4. A word is found that appears on the left of the first entity or on the right of the second entity, has a child which is a relation term and dominates the two entities. That child is considered as the key term.

10.2.2 Self-training algorithm

The first model is trained using the training set and the machine learning method explained in Section 10.2.1. To improve the performance of our system, a self-training process has been applied. Figure 10.3 outlines this process. This process starts with the provided labelled data and unlabelled data. The labelled data is used to train a model which is used to tag the unlabelled data. In most self-training algorithms the instances with the highest confidence level are selected to be added to the labelled data. However, in our application this methodology did not work. The most confident instances were tagged incorrectly. So we considered the following two measures to select the best unlabelled instances.

- The confidence level must be in an interval. It must be more than a threshold α and less than a specified value β.
- The predicted value of selected instances must be equal to their predicted value by the rule-based system.

[3]Collins' head finding rule [22] has been used.

Features	Description
Relation term	Root of the portion of the dependency tree connecting phenotype and genotype
Stemmed relation term	Stemmed by MALLET
Relative position of relation term	Whether it is before the first entity, after the second entity or between them
The relation term combined with the dependency relation	To consider the grammatical role of the relation term in the dependency path.
The relation term and its related position	
Key term	described in Section 10.2.1
Key term and its related position	
Collapsed version of the dependency path	All occurrences of nsubj/nsubjpass are replaced with subj, rcmod/partmod with mod, prep x with x and everything else with O.
Second version of the collapsed dependency path	Only the prep_* of dependency relations are kept.
Negative dependency relation	A binary feature shows whether there is any node in the path between the entities which dominates a *neg* dependency relation. This feature is used to catch the negative relations.
prep_between	A binary feature checks for the existence of two consecutive prep_between links in a dependency path.

Table 10.1: List of dependency features

In each iteration at most a bounded number of instances are selected and added to the labelled data to prevent adding lots of incorrectly labelled data to the training set in the first iterations

Features	Description
Syntactic features	
Stemmed version of relation term in the LCA node of the two entities	If the head[3] of Least common ancestor (LCA) node of the two entities in the syntax tree is a relation term then this feature takes a stemmed version of the head word as its value, otherwise it takes a NULL value.
The label of each of the constituents in the path between the LCA and each entity combined with its distance from the LCA node	
Surface features	
Relation terms and their relative positions	The relation terms between two entities or in a short distance (4 tokens) from them.

Table 10.2: List of syntactic and surface features

when the model is not powerful enough to make good predictions. The number of instances selected according to these factors could be less than the upper bound.

As we know in some self-training algorithms (e.g. , SVM) choosing the most confident unlabelled instances and adding them to the labelled data causes overfitting [119]. We encountered a similar overfitting when we added the most confident unlabelled instances. Because the mistakes the algorithm makes in the first iterations propagate to the following iterations, the first measure was added to our method to prevent overfitting in our model.

We used relation extraction output from Ibn Faiz's rule-based tool as a constraint in the decision to add an unlabelled instance to the training set. The rule-based tool is implemented for protein-protein interactions. It does not have good performance on phenotype-genotype relation extraction. So, using this tool's advice along with the confidence level means that the relationship must be of a more general category than just phenotype-genotype relationships.

1. **Given:**
   - **A set L of labelled training examples**
   - **A set U of unlabelled examples**
   - **E: Maximum number of examples added in each iteration**
   - **Cut-off: The number of iterations**
   - **LCL: least confidence value**
   - **MCL: most confidence value**
2. **Label all examples in U by rule-based system**
3. **Loop for I iterations**
   - **Use L to train the classifier Ci and label the examples in U**
   - **Select E examples from U where their confidence level is more than LCL and less than MCL and their predicted value is equivalent to rule-based prediction**
   - **Add selected E examples to L and delete them from U**
4. **Loop for Cut-off−I iterations**
   - **Use L to train the classifier Ci and label the examples in U**
   - **Select E examples from U where their confidence level is more than LCL and less than MCL**
   - **Add selected E examples to L and delete them form U**

Figure 10.3: The self training process described in Section 10.2.2

However, at some point this constraint holds the system back from learning broader types of relationships in the phenotype-genotype category. Therefore this selection factor is used only for the first *i* iterations, and after *i* iterations the best unlabelled data is chosen only based on the confidence level. Again, here, the confidence level must be in an interval.

This proposed self-training algorithm has been tried with various configurations and each variable in this process has been given several values. Each resulting model has been tried separately with our test set and the best system is selected based on its performance on the test set. In our best configuration 15 unlabelled instances are added to the labelled data in each iteration, in the first 5 iterations predictions made by the rule-based tool are taken into account, the least confidence level is 85%, the highest confidence level is 92% and the process stops after six iterations.

10.3 Results and discussion

The proposed system has been evaluated using the separate test set manually annotated by a biochemistry graduate student. The distribution of our data (number of sentences and number of genotype-phenotype pairs in each set) is illustrated in Table 10.3. The number of positive instances and negative instances in the unlabelled data are not available.

Data Set	Sentences	Instances	Positive instances	Negative instances
Training set	509	845	576	269
Test set	244	823	536	287
Unlabelled data	408	823	NA	NA

Table 10.3: Distribution of data in our different sets.

Table 10.4 shows the results obtained by the supervised learning algorithm and the proposed self-training algorithm. The results of testing Ibn Faiz's [34] rule-based and machine learning based tools are included in the table. Although these tools were not designed to be used for such an application, it is interesting to see that we could use these tools as the basis of a new

system for a different yet related application.

Method	Precision	Recall	F-measure
Ibn Faiz's ML-based tool	75.19	53.17	62.29
Ibn Faiz's rule-based tool	77.77	38.04	51.09
Supervised learning method	76.47	77.61	77.03
Self-training method	77.40	77.98	77.69

Table 10.4: Evaluation results

As can be seen in the table, Ibn Faiz's tools especially the rule-based system have good precision. This, we believe, is why considering rule-based predictions as a factor for choosing the best predictions was useful. However, their recall are quite low as one would expect. These results mean that there are some forms that are common between protein-protein and phenotype-genotype interactions, and these forms are what can be detected by these two tools. However, there are some forms of phenotype-genotype interactions which are not covered by the knowledge contained in these tools.

As illustrated in the table, our attempt to use knowledge (rules in the rule-based system and features in the machine learning-based system) incorporated in these earlier systems for training a new system was successful. We obtained good performance by only making a small training set and then we are able to improve the results by using our proposed self-training algorithm.

As mentioned before (see Section 10.1), there are some issues related to sentences from the *Phenominer* corpus which make them different from other sentences in our data. Table 10.5 shows the results after deleting the *Phenominer* sentences from our test set. The results are improved. Our proposed system is more effective for working with phenotypes not at the cellular level.

While we tried to add some negative sentences to our data to make it more balanced, Table 10.3 shows that our data is still biased: the number of negative instances is less than the number of positive instances. A more balanced training set is likely to improve the performance of the

Method	Precision	Recall	F-measure
Supervised learning method	80.20	79.79	80.00
Self-training method	80.05	81.07	80.55

Table 10.5: Results after deleting *Phenominer* sentences from the test set.

trained model.

Although we used a self-training algorithm to increase the size of our training set, because the best results were reached only after six iterations, the last training set has only 935 instances. Adding more instances using previously mentioned factors only decreased the performance. Our suggestion is to add more manually annotated sentences to the training set, so that the first model made by this set makes better predictions with more reliable confidence level, then use the self-training algorithm to improve the results.

The annotated sentences in the test set can be used as a valuable source of information. Going through the comments provided by the expert annotator, we have gained some insights on which important elements should be considered in a future system. The results of analyzing these sentences may imply modifying the feature set in the machine-learning method. We especially note the relation terms indicated by our annotator. The relation terms are specific to phenotype-genotype relationships and can be used to enhance this feature.

Our system is not capable of differentiating between sentences which include a relationship and sentences which only address the main objective of the research being discussed and do not conclude any interaction between phenotypes and genotypes. Finding and ignoring such sentences could improve the results. Here are some examples of such sentences:

- A cross-sectional study to determine the relationship of serum Haptoglobin-12 concentration with glycated albumin-16 and hemoglobinA (1c)-21 concentrations was conducted .

- We investigated whether mutations in the PRSS1 gene-8 are associated with hereditary and non-hereditary pancreatitis-17.

- The objective of this study was to investigate gene polymorphisms of detoxification enzymes and to determine whether the enzyme concentration and activity of glutathione S transferase microliter 1-28 correlates with the genotype in patients with cancer-36 of the oral cavity .

When the sentence is more complicated the system fails to find the correct relations. For example in the following sentence *"Serum levels of anti-gp70 Abs-7 were closely correlated with the presence of renal disease-16, more so than anti-dsDNA Abs-24."* only the relationship between *anti-gp70 Abs-7* and *renal disease-16* are extracted but the more complicated relation between *renal disease-16* and *anti-dsDNA Abs-24* is missed. More examples of this type will be needed in the training set and possibly features capturing the necessary linguistic features will need to be added to the feature set.

Chapter 11

Conclusions and Future Work

Throughout this thesis we have talked about the importance of a specialized system for extracting phenotype-genotype interactions from biomedical text. Such a system extracts important information about the complex heritable diseases caused by genes which is useful to both patients and physicians and also can be used in biomedical database curation. Developing this system demands finding solutions for three different issues: genotype name recognition, phenotype name recognition and phenotype-genotype relation extraction.

Extracting gene and protein names from biomedical literature is a well-studied problem. Different methods for gene and protein name extraction have been proposed and many of them participated in the BioCreative [1] contest. We decided to use BANNER [62], an open source biomedical named entity recognition system which was trained and tested on the BioCreative corpus and achieved good results in comparison with other named entity recognition systems.

Phenotypes, unlike genotypes, are complex concepts and do not consist of a homogeneous class of objects. Therefore phenotype name recognition is a very complicated task. We were the first group [1] to start working on this problem. We proposed two different phenotype name recognition methods.

- The first method is a rule-enhanced dictionary-based method. This system uses MetaMap to map UMLS semantic types to phrases in text then applies five rules and makes use of

[1] Maryam Khordad, Dr. Robert E. Mercer, Dr. Peter Rogan

HPO to decide if a phrase is a phenotype or not.

- The second method is a machine learning-based method based on Conditional Random Fields (CRF). The rules acquired in the rule-based method were incorporated in this system using features and some specific processing steps. This system also makes use of MetaMap and HPO.

Developing the machine learning-based method required a corpus to train a model and test it. As we did not have access to any prepared data, we created a corpus using a self-training algorithm and HPO. Furthermore, a separate test set was annotated manually. The system was then evaluated using both a 10-fold cross validation and a separate test set. The results are quite promising.

Our phenotype name recognition system was extremely dependent on MetaMap and also made mistakes in case the head of an NP was among a list of empty heads. We proposed solutions to overcome these problems and improve the performance of the system. The effectiveness of these solutions has been discussed based on both loose and strict evaluation results.

The final part of the thesis was related to phenotype-genotype relation extraction. Again we started this task without having any available resources or any previous knowledge about the nature of these relations. A semi-automatic method was proposed to construct a small training set. Then a self-training algorithm was applied to build a machine-learning model which could be used in extracting phenotype-genotype relations. The final system was evaluated using a separate test set annotated manually by a biochemistry graduate student and the results confirmed its good performance.

To summarize, our contributions in this thesis are the following:

- Using MetaMap and the semantic group *Disorders* to extract phenotype names.

- Proposing five rules to enhance phenotype name recognition.

- Using conditional random fields for phenotype name recognition.

- Proposing new features to incorporate our previously proposed five rules in a machine learning based system.
- Proposing a self-training algorithm to create a corpus automatically for phenotype name recognition.
- Preparing a manually annotated test set for phenotype name recognition.
- Proposing the bracketing and empty head solutions to improve the phenotype name recognition results.
- Proposing a semi-automatic method for making a small training set using two available relation extraction tools.
- Using a maximum entropy classifier for phenotype-genotype relation extraction.
- Developing a self-training algorithm to enlarge the training set and improve the phenotype-genotype relation extraction results.

11.1 Future work

Our rule-based phenotype recognition system had some problems. We proposed a machine learning-based system to solve these problems and then discussed some solutions to improve the results of the machine learning-based system. However, there are still more possible improvements.

- Our corpus is annotated automatically so it might include several incorrect phenotypes while some other phenotypes might be missing. It would be good to select a statistically representative subset of the phenotype corpus and ask an expert to annotate the corpus manually to understand how complete and accurate the tagged corpus is.
- Changing the feature set and adding more features can be considered.

- It would be beneficial to try other machine learning methods and compare their results with what we achieved by using CRF.

- In our current training set the complete form of the phenotype name (phenotypes along with their adjective and adverb modifiers) are not annotated. That is why our attempt to improve the strict evaluation results was not so successful. Using the BLLIP parser results, it might be possible to change the annotation of the corpus and test set to include the modifiers of phenotypes to achieve better results in strict evaluation.

- The current test set was annotated by the author. I suggest asking an expert to annotate the corpus. In this way we could be more confident about the performance of the system and also we can find which types of phenotypes are confusing for even human beings and find solutions for extracting them correctly.

Our proposed method for phenotype-genotype relation extraction achieved good results. However, there is still room for improvement. The following are suggestions for the future work related to this system.

- The training set made for this task was small. Despite our efforts to use a self-training algorithm to enlarge the corpus, the final set is still small. Asking an expert to annotate more sentences to be added to the corpus is recommended.

- The distribution of the training set shows that it does not have enough negative instances. A significant number of sentences without any relations between phenotypes and genotypes must be added to the training set in order to balance it.

- We have a small test set which was annotated manually by an expert. Analyzing the sentences and the annotator's comments could give some ideas about the nature of existing relations between phenotypes and genotypes. This information can be used in the following ways:

  - To understand if the system needs more features and if so what kind of features are needed.

  - To recognize what types of relations exist between phenotypes and genotypes and to add enough sentences for each type to the training set.
  - To extract rules related to phenotype-genotype relations and make a hybrid system which integrates both a rule-based and a machine learning-based method.
  - Furthermore, the annotator tagged the verbs and terms that are used in the sentences to relate a phenotype and a genotype. A list of relational words can be generated based on these words and a feature can be added to show whether a word is in this list or not.

- Our system is not able to detect and ignore the sentences like *"We investigated whether mutations in the PRSS1 gene are associated with hereditary and non-hereditary pancreatitis."*. These sentences only mention the purpose of a study and do not confirm any relations, but our system finds relations in them which leads to false positives. The template of these sentences could be recognized and a pre-processing step could be incorporated in the system to disregard these types of sentences before from being considered as a potential source of phenotype-genotype relationships.

- The current system is not able to extract complicated relations where a pronoun refers to a phenotype or a genotype in the same sentence or the previous sentences (anaphora) or where a part of or the whole genotype or phenotype is omitted (ellipsis) in a sentence. In the case of anaphora, coreference resolution techniques can be used to find the objects referred to by the pronouns. Finding and resolving ellipsis is a difficult task because of the missing text elements. Winograd [113] discusses methods proposed for ellipsis resolution.

- It is worth trying other machine learning methods, like Support Vector Machines which have achieved good results in extracting interactions in other applications [13], for this problem and compare the results with our current method.

Bibliography

[1] Biocreative. http://www.biocreative.org/. Online; accessed 1-March-2014.

[2] Genbank. http://www.ncbi.nlm.nih.gov/genbank/. Online; accessed 1-March-2014.

[3] Pubmed. http://www.ncbi.nlm.nih.gov/pubmed. Online; accessed 1-March-2014.

[4] Steven Abney. Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 360–367, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[5] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped blast and psiblast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[6] M. A. Andrade and A. Valencia. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7):600–607, 1998.

[7] Alan R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *AMIA Symposium*, pages 17–21, 2001.

[8] M. Ashburner. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[9] Amos Bairoch and Rolf Apweiler. The swiss-prot protein sequence data bank and its supplement trembl in 1999. *Nucleic Acids Research*, 27(1):49–54, 1999.

[10] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled sata with co-training. In Peter L. Bartlett and Yishay Mansour, editors, *COLT*, pages 92–100. ACM, 1998.

[11] Phil Blunsom and Trevor Cohn. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 65–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[12] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, August 1996.

[13] Quoc-Chinh Bui, Sophia Katrenko, and Peter M. A. Sloot. A hybrid approach to extract protein-protein interactions. *Bioinformatics*, 27(2):259–265, 2011.

[14] Anita Burgun, Fleur Mougin, and Olivier Bodenreider. Two approaches to integrating phenotype and clinical information. *AMIA Annual Symposium proceedings*, 2009:75–79, 2009.

[15] Bob Carpenter and Breck Baldwin. *Natural Language Processing with LingPipe 4*. LingPipe Publishing, New York, draft edition, June 2011.

[16] Jeffrey T. Chang, Hinrich Schütze, and Russ B. Altman. GAPSCORE: finding gene and protein names one word at a time. *Bioinformatics (Oxford, England)*, 20(2):216–225, January 2004.

[17] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 173–180, 2005.

[18] Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 355–362, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[19] Stephen Clark, James R. Curran, and Miles Osborne. Bootstrapping POS taggers using unlabelled data. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 49–55, Stroudsburg, PA, USA, 2003.

[20] Peter Clifford. Markov random fields in statistics. In Geoffrey Grimmett and Dominic Welsh, editors, *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, pages 19–32. Oxford University Press, Oxford, 1990.

[21] Nigel Collier, Mai-Vu Tran, Hoang-Quynh Le, Anika Oellrich, Ai Kawazoe, Martin Hall-May, and Dietrich Rebholz-Schuhmann. A hybrid approach to finding phenotype candidates in genetic texts. In *Proceedings of COLING 2012*, pages 647–662, Mumbai, India, December 2012.

[22] Michael Collins. Head-driven statistical models for natural language parsing. *Comput. Linguist.*, 29(4):589–637, December 2003.

[23] Gordon V. Cormack. Harnessing unlabeled examples through iterative application of dynamic Markov modeling. In *In Proceedings of the ECML-PKDD Discovery Challenge Workshop*, 2006.

[24] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[25] Adrien Coulet, Nigam H. Shah, Yael Garten, Mark A. Musen, and Russ B. Altman. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics*, 43(6):1009–1019, 2010.

[26] Mark Craven. Learning to extract relations from medline. In *AAAI-99 Workshop on Machine Learning for Information Extraction*, pages 25–30, 1999.

[27] Aron Culotta, Ron Bekkerman, and Andrew McCallum. Extracting social networks and contact information from email and the Web. In *Conference on Email and Anti-Spam 1*, 2004.

[28] M. Dai, N.H. Shah, W. Xuan, MA. Musen, S.J. Watson, B.D. Athey, and F. Meng. An Efficient Solution for Mapping Free Text to Ontology Terms. *AMIA Summit on Translational Bioinformatics, San Francisco, CA*, 2008.

[29] John Day-Richter, Midori A. Harris, Melissa Haendel, The Gene Ontology OBO, and Suzanna Lewis. OBO-Edit an ontology editor for biologists. *Bioinformatics*, 23(16):2198–2200, August 2007.

[30] Marie-Catherine de Marneffe and Christopher D. Manning. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[31] Jing Ding, Daniel Berleant, Dan Nettleton, and Eve Syrkin Wurtele. Mining medline: Abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing*, pages 326–337, 2002.

[32] Harold J Drabkin, Christopher Hollenbeck, David P Hill, and Judith A Blake. Ontological visualization of protein-protein interactions. *BMC Bioinformatics*, 6(1):1–11, 2005.

[33] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[34] Mohammad Syeed Ibn Faiz. Discovering higher order relations from biomedical text. Master's thesis, University of Western Ontario, London, ON, Canada, 2012.

[35] Kristofer Franzén, Gunnar Eriksson, Fredrik Olsson, Lars Asker, Per Lidén, and Joakim Cöster. Protein names and how to find them. *International journal of medical informatics*, 67(1-3):49–61, December 2002.

[36] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics (Oxford, England)*, 17 Suppl 1(suppl_1):S74–82, June 2001.

[37] 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.

[38] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relex - relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[39] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett. Protein structures and information extraction from biological texts: the PASTA system. *Bioinformatics*, 19(1):135–143, January 2003.

[40] William M. Gelbart, Madeline A. Crosby, B. Matthews, W. P. Rindone, J. Chillemi, S. Russo Twombly, David Emmert, Michael Ashburner, Rachel A. Drysdale, E. Whitfield, Gillian H. Millburn, A. de Grey, T. Kaufman, K. Matthews, David Gilbert, Victor B. Strelets, and C. Tolstoshev. Flybase: A Drosophila database. *Nucleic Acids Research*, 25(1):63–66, 1997.

[41] Louise Guthrie, Brian M. Slator, Yorick Wilks, and Rebecca Bruce. Is there content in empty heads? In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90, pages 138–143, Stroudsburg, PA, USA, 1990. Association for Computational Linguistics.

[42] Jörg Hakenberg, Steffen Bickel, Conrad Plake, Ulf Brefeld, Hagen Zahn, Lukas Faulstich, Ulf Leser, and Tobias Scheffer. Systematic feature evaluation for gene name recognition. *BMC bioinformatics*, 6 Suppl 1(Suppl 1), 2005.

[43] J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.

[44] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S14, 2005.

[45] Xiaofen He and Chrysanne DiMarco. Using lexical chaining to rank protein-protein interactions in biomedical texts. In *BioLink 2005: Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Conference of the Association for Computational Linguistics (poster Presentation)*, 2005.

[46] Florence Horn, Anthony L. Lau, and Fred E. Cohen. Automated extraction of mutation data from the literature: application of MuteXt to G protein-coupled receptors and nuclear hormone receptors. *Bioinformatics*, 20(4):557–568, Mar 2004.

[47] D. Hristovski, C. Friedman, T. C. Rindflesch, and B. Peterlin. Exploiting semantic relations for literature-based discovery. *AMIA Annual Symposium proceedings*, pages 349–353, 2006.

[48] Dimitar Hristovski, Borut Peterlin, Joyce A. Mitchell, and Susanne M. Humphrey. Using literature-based discovery to identify disease candidate genes. *I. J. Medical Informatics*, 74(2-4):289–298, 2005.

[49] Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G. Payan, Kunbin Qu, and Ming Li. Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612, December 2004.

[50] B. L. Humphreys, D. A. Lindberg, H. M. Schoolman, and G. O. Barnett. The Unified Medical Language System: an informatics research collaboration. *J Am Med Inform Assoc*, 5(1):1–11, 1998.

[51] K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. *Pacific Symposium on Biocomputing*, pages 505–516, 2000.

[52] TK Jenssen, A Laegreid, J Komorowski, and E Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28, May 2001.

[53] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[54] Sophia Katrenko and Pieter Adriaans. Learning relations from biomedical corpora using dependency trees. In *Knowledge Discovery and Emergent Complexity in Bioinformatics*, volume 4366 of *Lecture Notes in Computer Science*, pages 61–80. Springer Berlin / Heidelberg, 2007.

[55] Maryam Khordad, Robert E. Mercer, and Peter Rogan. Improving phenotype name recognition. *Canadian Conference on AI*, 6657:246–257, 2011.

[56] Maryam Khordad, Robert E Mercer, and Peter Rogan. A machine learning approach for phenotype name recognition. In *Proceedings of COLING 2012*, pages 1425–1440, Mumbai, India, December 2012.

[57] T. E. Klein, J. T. Chang, M. K. Cho, K. L. Easton, R. Fergerson, M. Hewett, Z. Lin, Y. Liu, S. Liu, D. E. Oliver, D. L. Rubin, F. Shafa, J. M. Stuart, and R. B. Altman. Integrating genotype and phenotype information: an overview of the PharmGKB project. *The Pharmacogenomics Journal*, 1(3):167–170, 2001.

[58] 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.

[59] 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.

[60] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying Conditional Random Fields to Japanese Morphological Analysis. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 230–237, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[61] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[62] Robert Leaman and Graciela Gonzalez. Banner: An executable survey of advances in biomedical named entity recognition. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Tiffany Murray, and Teri E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 652–663. World Scientific, 2008.

[63] Gondy Leroy, Hsinchun Chen, and Jesse D. Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of Biomedical Informatics*, 36(3):145–158, 2003.

[64] Gondy Leroy, Hsinchun Chen, and Jesse D. Martinez. A shallow parser based on closed-class words to capture relations in biomedical text. *J. of Biomedical Informatics*, 36(3):145–158, June 2003.

[65] Ulf Leser and Jörg Hakenberg. What makes a gene name? named entity recognition in the biomedical literature. *Briefings in Bioinformatics*, 6(4):357–369, 2005.

[66] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, February 1994.

[67] Edward M. Marcotte, Ioannis Xenarios, and David Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17(4):359–363, 2001.

[68] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.

[69] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[70] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[71] David McClosky and Eugene Charniak. Self-training for biomedical parsing. In *Proceedings of the Association for Computational Linguistics (ACL 2008, short papers)*, pages 101–104, Columbus, Ohio, 2008. The Association for Computer Linguistics.

[72] AT McCray, A Burgun, and O Bodenreider. Aggregating UMLS semantic types for reducing conceptual complexity. *Proceedings of Medinfo*, 10(pt 1):216–20, 2001.

[73] V. McKusick. Mendelian Inheritance in Man and Its Online Version, OMIM. *The American Journal of Human Genetics*, 80(4):588–604, April 2007.

[74] Rada Mihalcea. Co-training and self-training for Word Sense Disambiguation. In Hwee, editor, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning*, pages 33–40, May 2004.

[75] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[76] See-kiong Ng and Marie Wong. Toward routine automatic pathway discovery from online scientific text abstracts. *Genome Informatics*, 10:104–112, 1999.

[77] Vincent Ng and Claire Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 94–101, Stroudsburg, PA, USA, 2003.

[78] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, pages 86–93, New York, NY, USA, 2000. ACM.

[79] Chikashi Nobata, Nigel Collier, and Jun ichi Tsujii. Automatic term identification and classification in biology texts. In *In Proc. of the 5th NLPRS*, pages 369–374, 1999.

[80] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *In Proceedings of the Human Language Technology Conference*, pages 73–77, 2002.

[81] Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[82] Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, pages 329–336, 2004.

[83] David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 235–242, New York, NY, USA, 2003. ACM.

[84] Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Bjorne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6+, 2008.

[85] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[86] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286. IEEE, February 1989.

[87] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[88] Thomas C. Rindflesch, Bisharah Libbus, Dimitar Hristovski, Alan R. Aronson, and Halil Kilicoglu. Semantic relations asserting the etiology of genetic diseases. *AMIA Annual Symposium Proceedings*, pages 554–558, 2003.

[89] Thomas C. Rindflesch, Lorraine Tanabe, John N. Weinstein, and Lawrence Hunter. Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Pacific Symposium on Biocomputing*, volume 5, pages 514–525, 2000.

[90] P. N. Robinson and S. Mundlos. The human phenotype ontology. *Clinical genetics*, 77(6):525–534, June 2010.

[91] Dan Roth and Wen-tau Yih. Integer linear programming inference for conditional random fields. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 736–743, New York, NY, USA, 2005. ACM.

[92] Anoop Sarkar. Applying co-training methods to statistical parsing. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational*

Linguistics on Language technologies, NAACL '01, pages 1–8, Stroudsburg, PA, USA, 2001.

[93] Ariel S. Schwartz and Marti A. Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing*, pages 451–462, 2003.

[94] Isabel Segura-Bedmar, Paloma Martinez, and Maria Segura-Bedmar. Drug name recognition and classification in biomedical texts: A case study outlining approaches underpinning automated systems. *Drug Discovery Today*, 13(17-18):816 – 823, 2008.

[95] T. Sekimizu, H. S. Park, and J. Tsujii. Identifying the interaction between genes and gene products based on frequently seen verbs in MEDLINE abstracts. *Genome Informatics*, pages 62–71, 1998.

[96] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA)*, pages 104–107, 2004.

[97] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 134–141, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[98] H. Shatkay and R. Feldman. Mining the biomedical literature in the genomic era: an overview. *J Comput Biol*, 10(6):821–855, 2003.

[99] Cynthia Smith, Carroll A. Goldsmith, and Janan Eppig. The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology*, 6(1):R7+, 2004.

[100] L. Smith, T. Rindflesch, and W. J. Wilbur. MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics (Oxford, England)*, 20(14):2320–2321, September 2004.

[101] B. J. Stapley and G. Benoit. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in medline abstracts. In *Pac. Symp. Biocomput*, pages 529–540, 2000.

[102] M. Stephens, M. Palakal, S. Mukhopadhyay, R. Raje, and J. Mostafa. Detecting gene relations from Medline abstracts. *Pac Symp Biocomput*, pages 483–495, 2001.

[103] Tom Strachan and Andrew Read. *Human Molecular Genetics, Third Edition*. Garland Science/Taylor & Francis Group, November 2003.

[104] Charles A. Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007.

[105] D.R. Swanson. Fish oil, raynauds syndrome, and undiscovered public knowledge. *Perspect. Bio. Med*, 30:7–18, 1986.

[106] L. Tanabe, U. Scherf, L. H. Smith, J. K. Lee, L. Hunter, and J. N. Weinstein. MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling. *BioTechniques*, 27(6):1210–1217, December 1999.

[107] Lorraine Tanabe and W. John Wilbur. Tagging gene and protein names in full text articles. In *Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 9–13, Phildadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

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

[109] Alfonso Valencia. Automatic annotation of protein function. *Current opinion in structural biology*, 15(3):267–274, June 2005.

[110] Hester M. Wain, Michael J. Lush, Fabrice Ducluzeau, Varsha K. Khodiyar, and Sue Povey. Genew: the human gene nomenclature database, 2004 updates. *Nucleic Acids Research*, 32(Database-Issue):255–257, 2004.

[111] Hanna M. Wallach. Conditional random fields: An introduction. Technical report, Rapport technique MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.

[112] Michael Wick, Khashayar Rohanimanesh, Andrew Mccallum, and Anhai Doan. A Discriminative Approach to Ontology Alignment. In *International Workshop on New Trends in Information Integration (NTII) at the conference for Very Large Databases (VLDB WS)*, 2008.

[113] T. Winograd. *Language as a Cognitive Process Volume 1: Syntax*. Addison-Wesley, Reading, MA, 1983.

[114] R. Xu, K. Supekar, A. Morgan, A. Das, and A. Garber. Unsupervised method for automatic construction of a disease dictionary from a large free text collection. *AMIA Annual Symposium proceedings*, pages 820–824, 2008.

[115] A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii. Event extraction from biomedical papers using a full parser. *Pacific Symposium on Biocomputing*, pages 408–419, 2001.

[116] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

[117] Hao Yu, Xiaoyan Zhu, Minlie Huang, and Ming Li. Discovering patterns to extract protein-protein interactions from the literature: Part ii. *Bioinformatics*, 21(15):3294–3300, 2005.

[118] Deyu Zhou and Yulan He. Extracting interactions between proteins from the literature. *Journal of biomedical informatics*, 41(2):393–407, April 2008.

[119] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

Appendix A

A partial list of UMLS semantic types and semantic groups

Semantic Group	Semantic Type
Anatomy	Anatomical Structure
	Body Location or Region
	Body Part, Organ, or Organ Component
	Body Space or Junction
	Body Substance
	Body System
	Cell
	Cell Component
	Embryonic Structure
	Fully Formed Anatomical Structure
	Tissue

Semantic Group	Semantic Type
Disorders	Acquired Abnormality
	Anatomical Abnormality
	Cell or Molecular Dysfunction
	Congenital Abnormality
	Disease or Syndrome
	Experimental Model of Disease
	Finding
	Injury or Poisoning
	Mental or Behavioral Dysfunction
	Neoplastic Process
	Pathologic Function
	Sign or Symptom
Physiology	Cell Function
	Clinical Attribute
	Genetic Function
	Mental Process
	Molecular Function
	Organism Attribute
	Organism Function
	Organ or Tissue Function
	Physiologic Function

Appendix B

List of special modifiers

1. abnormal
2. decreased
3. increased
4. reduced
5. absent
6. small
7. absence of
8. enlarged
9. short
10. dilated
11. impaired
12. thin
13. hypoplastic

14. delayed
15. elevated
16. failure of
17. loss of
18. low
19. fused
20. altered
21. thick
22. pancreatic
23. hyperplastic
24. disorganized
25. kidney
26. uterine
27. renal
28. long
29. high
30. rudimentary
31. muscle
32. intestinal
33. enhanced

34. retinal
35. eye
36. embryonic
37. uterus
38. liver
39. truncated
40. large
41. hepatic
42. defective
43. prolonged
44. skin
45. pulmonary
46. lung
47. ectopic
48. vascular
49. immune
50. atrophic
51. heart
52. cardiac
53. white

54. right
55. neurological
56. behavioral
57. mammary
58. t
59. respiratory
60. premature
61. male
62. early
63. bifid
64. urinary
65. skeletal
66. pancreas
67. ovarian
68. optic
69. ocular
70. hair
71. excessive
72. blood
73. b

74. thickened
75. spinal
76. pituitary
77. lip
78. gastric
79. esophageal
80. degeneration of
81. cervical
82. central
83. aortic
84. ventricular
85. prostate
86. irregular
87. heterotopic
88. eyelid
89. detached
90. cranial
91. chronic
92. choroid
93. brain

94. adrenal
95. vitreous
96. vaginal
97. type
98. tail
99. survival
100. spleen
101. reproductive
102. underdeveloped
103. hypoplasia of

Appendix C

List of relation verbs demonstrating relationships between genotypes and phenotypes [88]

1. account
2. cause
3. effect
4. evoke
5. generate
6. induce
7. lead
8. produce
9. provoke
10. result

11. trigger
12. contribute
13. predispose
14. promote
15. predict
16. associate
17. characterise
18. characterize
19. correlate
20. decrease
21. involve
22. influence
23. link
24. mediate
25. occur
26. relate
27. restore

Appendix D

Relational terms used by Ibn Faiz's rule-based system [34]

abolish, abolished, abolishes, abolishing, abrogat, acceler, accelerat, acceptor, accompanied, accompanies, accompany, accompanying, accumul, accumulation, acetylat, acetylate, acetylated, acetylates, acetylating, acetylation, acquir, act, acting, action, activ, activat, activate, activated, activates, activating, activation, activator, acts, adapt, add, addit, adhe, adher, affect, affects, affinities, affinity, aggregat, agoni, agonist, alter, altered, altering, amplif, antagoni, apparat, assembl, assist, associat, associate, associated, associates, associating, association, associations, attach, attached, attaches, attaching, attachment, attack, attacked, attacking, attacks, attenuat, attenuate, attenuated, attenuates, attenuating, augment, augmented, augmenting, augments, autophosphorylat, autoregulat, bind, binding, binds, block, blockage, blocked, blocking, blocks, bound, carbamoylated, carbamoylation, carboxyl, carboxylate, carboxylates, carboxylation, cataly, cause, caused, causes, causing, change, changed, changes, changing, characterization, characterized, cleav, cleavage, cleave, cleaved, cleaves, cleaving, cluster, co-expression, co-immunoprecipitate, co-immunoprecipitated, co-immunoprecipitates, co-immunoprecipitating, co-immunoprecipitation, co-immunoprecipitations, co-localization, co-localized, co-localizing, co-operat, co-precipit, co-precipitate, co-precipitated, co-precipitates, co-precipitating, co-precipitation, co-precipitations, co-purifi, co-stimulate, co-stimulated, co-

stimulating, coactivat, coactivator, coassociation, coexist, coexpres, coexpression, coimmunoprecipitate, coimmunoprecipitated, coimmunoprecipitates, coimmunoprecipitating, coimmunoprecipitation, coimmunoprecipitations, colocaliz, colocalization, colocalized, compet, compete, competed, competes, competing, complex, complexation, complexed, complexes, complexing, component, compris, concentration, conjugat, conjugate, conjugated, conjugates, conjugating, conjugation, conserved, consisted, consisting, consists, contact, contacted, contacting, contacts, contain, contained, containing, contains, contribute, contributed, contributes, contributing, control, controled, controling, controlled, controlling, controls, convers, convert, converted, converting, converts, cooperat, cooperate, cooperated, cooperates, cooperating, cooperative, coprecipit, coprecipitate, coprecipitated, coprecipitates, coprecipitating, copurifi, correlat, correlate, correlated, correlating, correlation, costimulate, costimulated, costimulating, counteract, counterreceptor, coupl, cripple, crippled, cripples, crippling, cross-link, cross-linked, cross-linking, cross-links, cross-react, cross-reacted, cross-reacting, cross-reacts, cross-talk, crosslink, crosslinker, crosslinking, crosstalk, deacetylat, deacetylate, deacetylated, deacetylates, deacetylating, deacetylation, deaminated, deamination, decarboxylated, decarboxylates, decarboxylation, declin, decreas, decrease, decreased, decreases, decreasing, degrad, degrade, degraded, degrades, degrading, dehydrated, dehydrogenated, dehydrogenation, depend, depended, dependent, depending, depends, dephosphorylat, dephosphorylate, dephosphorylated, dephosphorylates, dephosphorylating, dephosphorylation, deplet, deposi, depress, depressed, depresses, depressing, deriv, destruct, determine, determined, determines, determining, dimer, diminish, diminished, diminishes, diminishing, direct, directed, directing, directs, disrupt, disrupted, disrupting, disruption, disrupts, dissociat, dissociate, dissociated, dissociating, dissociation, distribute, distributed, distributes, distribution, dock, docked, docking, docks, down-regulat, down-regulate, down-regulated, down-regulates, down-regulating, down-regulation, downregulat, downregulate, downregulated, downregulates, downregulating, downregulation, drive, driven, drives, driving, effect, effected, effecting, effects, elavating, elevat, elevate, elevated, elevates, elevating, eliminate, eliminated, eliminates, eliminating, encod, encode, encoded, encodes, encoding, engage, engaged, engages, engaging, enhanc, enhance, en-

hanced, enhances, enhancing, enrich, evoke, evoked, exert, exhibit, expos, express, expressed, expresses, expressing, expression, facilitate, facilitates, facilitating, faciliteted, follow, followed, following, follows, form, formation, formed, forms, formylated, functio, function, functioned, functions, fuse, fused, fuses, fusing, generat, generate, generated, generates, generating, glucosyl, glycosyl, glycosylated, glycosylates, glycosylation, govern, governed, governing, governs, heterodimer, heterodimerization, heterodimerize, heterodimerized, heterodimerizes, heterodimerizing, heterodimers, homodimer, homodimerization, homodimerize, homodimerized, homodimerizes, homodimers, homologous, homologue, hydrol, hydrolyse, hydrolysed, hydrolyses, hydrolysing, hydrolysis, hyperexpr, identified, imitat, immuno-precipit, immunoprecipit, immunoprecipitate, immunoprecipitated, immunoprecipitates, immunoprecipitating, impact, impacted, impacting, impacts, impair, impaired, impairing, impairs, implicate, implicated, import, improv, inactivat, inactivate, inactivated, inactivates, inactivating, inactivation, inactive, includ, incorporate, incorporated, incorporates, incorporation, increas, increase, increased, increases, increasing, increment, induc, induce, induced, induces, inducing, induction, influenc, influence, influenced, influences, influencing, inhibit, inhibited, inhibiting, inhibition, inhibitor, inhibits, initiat, initiate, initiated, initiates, initiating, interact, interacted, interacting, interaction, interactions, interacts, interfer, interrupt, involve, involved, involvement, involves, involving, isomerization, isomerize, isomerized, isomerizes, isomerizing, lead, leading, leads, led, ligand, ligate, ligated, ligates, ligating, ligation, limit, limited, limiting, limits, link, linked, linking, links, localization, mediat, mediate, mediated, mediates, mediating, methylate, methylated, methylates, methylating, methylation, migrat, mobili, mobilisation, mobilise, mobilised, mobilises, mobilising, mobilization, mobilize, mobilized, mobilizes, mobilizing, moderat, modif, modified, modifies, modify, modifying, modulat, modulate, modulated, modulates, modulating, neutrali, neutralise, neutralised, neutralises, neutralising, neutralize, neutralized, neutralizes, neutralizing, obstruct, operat, oppos, overexpress, overproduc, oxidis, oxidiz, oxidization, oxidize, oxidized, oxidizes, oxidizing, pair, paired, pairing, pairs, peroxidizing, perturb, perturbed, perturbing, perturbs, phosphoryates, phosphorylat, phosphorylate, phosphorylated, phosphorylates, phosphorylating, phosphorylation, potentiat, potentiate, potentiated, po-

tentiates, potentiating, prducing, precede, preceded, precedes, preceding, prevent, prevented, preventing, prevents, process, produc, produce, produced, produces, producing, prohibit, promot, promote, promoted, promotes, promoting, raise, raised, raises, raising, react, reactivate, reactivated, reactivates, reactivating, recogni, recognise, recognised, recognises, recognising, recognize, recognized, recognizes, recognizing, recruit, recruited, recruiting, recruitment, recruits, reduc, reduce, reduced, reduces, reducing, reduction, regulat, regulate, regulated, regulates, regulating, regulation, regulator, relate, related, releas, remov, replac, repress, repressed, represses, repressing, requir, require, required, requires, requiring, respond, responded, responding, responds, respons, response, responses, responsible, result, resulted, resulting, results, reversed, secret, sequester, sequestered, sequestering, sequesters, sever, signal, signaled, signaling, signals, splice, stabili, stabilization, stabilized, stimulat, stimulate, stimulated, stimulates, stimulating, stimulation, subunit, suppress, suppressed, suppresses, suppressing, suspend, synergise, synergised, synergises, synergising, synergize, synergized, synergizes, synergizing, synthesis, target, targeted, targeting, targets, terminate, terminated, terminates, terminating, tether, tethered, tethering, tethers, trans-activate, trans-activated, trans-activates, trans-activating, transactivat, transactivate, transactivated, transactivates, transactivating, transamination, transcri, transcribe, transcribed, transcribes, transcribing, transduc, transform, transformed, transforming, transforms, translat, translocat, transport, transregulat, trigger, triggered, triggering, triggers, ubiquitinate, ubiquitinated, ubiquitinates, ubiquitinating, ubiquitination, up-regulat, up-regulate, up-regulated, up-regulates, up-regulating, up-regulation, upregulat, up-regulate, upregulated, upregulates, upregulating, upregulation, use, utilis, utiliz, yield, clone, cloning

Appendix E

Relational terms annotated by our annotator

- contribute
- associated
- implicates
- regulation
- overexpressed
- downregulation
- responsible
- down regulated
- association
- related
- relate to

- development
- increased
- caused
- identified
- decreased
- contain
- between
- linked
- correlation
- causes
- observed
- arises
- lead
- characterized
- resulting
- accompanied
- derived
- present
- correlated
- play

- explained
- depends
- induce
- cause
- increase
- moderate
- determine the relationship between
- presented
- predictor
- refers
- is (linked)
- evidenced
- confirmed
- may be
- determine (the role of)
- showed
- includes
- suggests
- were (predictors)
- include

- confers
- are (responsible)
- associations of
- resulted
- influence
- have (been identified)
- (correlation) was observed
- found
- contribution
- contributes
- results
- predispose
- affect
- account
- revealed
- involve
- causative
- inhibited
- displayed
- map

- evaluated
- screened

Appendix F

List of abbreviations

Abbreviation	Meaning
PPI	protein-protein interaction
UMLS	Unified Medical Language System
PharmGKB	Pharmacogenetics Knowledge Base
OMIM	Online Mendelian Inheritance in Man
HPO	Human Phenotype Ontology
SG	Semantic Group
ST	semantic Type
MPO	Mammalian Phenotype Ontology
NLP	Natural Language Processing
CRF	Conditional Random Fields
NLM	National Library of Medicine
WSD	word sense disambiguation
Mallet	MAchine Learning for LanguagE Toolkit
HMMs	Hidden Markov Models
PAC	Probably Approximately Correct

Abbreviation	Meaning
NP	noun phrases
CRF	conditional random fields
NER	Named Entity Recognition
POS	part of speech
HUGO	Human Genome Nomenclature
GO	Gene Ontology
MAPK	MAP kinase
FSA	finite state automata
POS	part-of-speech
ML	machine learning

Curriculum Vitae

Name:	Maryam Khordad
Post-Secondary Education and Degrees:	Shahid Beheshti University Tehran, Iran 2000 - 2005 BS Sharif University of Technology Tehran, Iran 2005 - 2007 MS University of Western Ontario London, ON 2009 - 2014 Ph.D.
Honours and Awards:	Graduate Thesis Research Award 2013-2014
Related Work Experience:	Teaching Assistant The University of Western Ontario 2009 - 2014

Publications:

Khordad, M., Mercer, R.E., Rogan, P.: A machine learning approach for phenotype name recognition. COLING 2012, 1425-1440

Khordad, M., Mercer, R.E., Rogan, P.: Improving phenotype name recognition. Canadian Conference on AI 2011, 246-257

Khordad, M., Maleki, M.: K Nearest Neighbors Based on Lateration for WLAN Location Estimation, IEEE, IET International Symposium on COMMUNICATION SYSTEMS, NETWORKS AND DIGITAL SIGNAL PROCESSING, 2010, 301-305.

Khordad, M., Farahani, Y., Sharif, L.: A Path Planning Solution in Mobile Robots with Minimum Sharp Variations of Controlled Parameters, 12th Annual Int. CSI Computer Conf., 2007, 2282-2287.

Khordad, M., Kazemeyni, F., ShamsFard, M.: A Hybrid Method to Categorize HTML Documents, 6th Int. Conf. on Data Mining (DATA MINING VI), 2005, 331-340.

Khordad, M., Kazemeyni, F., ShamsFard, M.: Conceptual Classification of Web Pages using Ontology, 10th Annual Int. CSI Computer Conf., 2005, 404-411.