August 2017

Unsupervised Biomedical Named Entity Recognition

Omid Ghiasvand
University of Wisconsin-Milwaukee

UNSUPERVISED BIOMEDICAL NAMED ENTITY RECOGNITION

by

Omid Ghiasvand

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Biomedical and Health Informatics

at

The University of Wisconsin-Milwaukee

August 2017

ABSTRACT

UNSUPERVISED BIOMEDICAL NAMED ENTITY RECOGNITION

by

Omid Ghiasvand

The University of Wisconsin-Milwaukee, 2017
Under the Supervision of Dr. Rohit J. Kate

Named entity recognition (NER) from text is an important task for several applications, including in the biomedical domain. Supervised machine learning based systems have been the most successful on NER task, however, they require correct annotations in large quantities for training. Annotating text manually is very labor intensive and also needs domain expertise. The purpose of this research is to reduce human annotation effort and to decrease cost of annotation for building NER systems in the biomedical domain. The method developed in this work is based on leveraging the availability of resources like UMLS (Unified Medical Language System), that contain a list of biomedical entities and a large unannotated corpus to build an unsupervised NER system that does not require any manual annotations.

The method that we developed in this research has two phases. In the first phase, a biomedical corpus is automatically annotated with some named entities using UMLS through unambiguous exact matching which we call weakly-labeled data. In this data, positive examples are the entities in the text that exactly match in UMLS and have only one semantic type which belongs to the desired entity class to be extracted (for example, diseases and disorders). Negative examples are the entities in the text that exactly match in UMLS but are of semantic types other than those that belong to the desired entity class. These examples are then used to train a machine learning classifier using features that represent the contexts in which they appeared in the text. The trained classifier is applied back to the text to gather more examples iteratively through the process of self-training. The trained classifier is then capable of classifying mentions in an unseen text as of the desired entity class or not from the contexts in which they appear.

Although the trained named entity detector is good at detecting the presence of entities of the desired class in text, it cannot determine their correct boundaries. In the second phase of our method, called “Boundary Expansion”, the correct boundaries of the entities are determined. This method is based on a novel idea that utilizes machine learning and UMLS. Training examples for boundary expansion are gathered directly from UMLS and do not require any manual annotations. We also developed a new WordNet based approach for boundary expansion.

Our developed method was evaluated on three datasets - SemEval 2014 Task 7 dataset that has diseases and disorders as the desired entity class, GENIA dataset that has proteins, DNAs, RNAs, cell types, and cell lines as the desired entity classes, and i2b2 dataset that has problems, tests, and treatments as the desired entity classes. Our method performed well and obtained performance close to supervised methods on the SemEval dataset. On the other datasets, it outperformed an existing unsupervised method on most entity classes. Availability of a list of entity names with their semantic types and a large unannotated corpus are the only requirements of our method to work well. Given these, our method generalizes across different types of entities and different types of biomedical text. Being unsupervised, the method can be easily applied to new NER tasks without needing costly annotations.

To my beloved parents,

without whom none of my success would be possible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Problem Statement and Motivation

Named entity recognition (NER) is a task of automatically identifying entities of certain types from text documents [1]; for example, identifying all gene names from biomedical literature [2], or identifying all person and organization names from news stories [3], or identifying all biomedical names from clinical text [4]. Figure 1 shows an example of the last task. NER has several important applications. It can help in identifying and highlighting passages of text that may be relevant for a particular information need [5]. It is used to index documents according to entity types in order to help in retrieving documents as needed, for example, to retrieve all documents from a collection that talk about a specific gene [6]. It is also used as the first step in automatically identifying relations between entities mentioned in text, a task also known as relation extraction [7]. For example, in order to automatically identify gene-gene interactions mentioned in text [8], a system first needs to automatically identify all the genes mentioned in the text. NER and relation extraction tasks are also referred to as information extraction [9-10] and have important applications in automatically populating databases [11]. Thus, NER is also useful in transforming unstructured representation of text into a structured knowledge representation of databases.

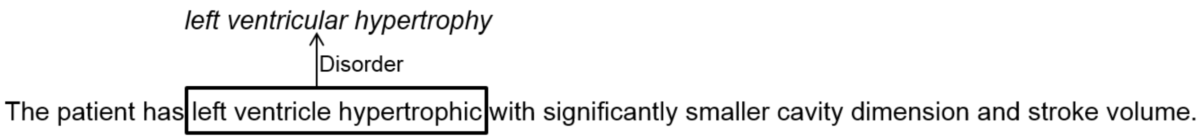

left ventricular hypertrophy
Disorder
The patient has left ventricle hypertrophic with significantly smaller cavity dimension and stroke volume.

Figure 1 An illustration of named entity recognition (NER) task of identifying disease and disorder entities from clinical text. The figure shows an example of clinical sentence from which a disorder entity has been recognized

NER is not as simple as matching known entity names in the text documents. There are two major reasons for this. The first reason is *variability*, i.e. entities may be mentioned in the text in various forms which are different from their standard names. For example, in Figure 1 the disorder name is “left ventricular hypertrophy” but it is mentioned in the sentence as “left ventricle hypertrophic” hence a match would have failed. The second reason is *ambiguity*, i.e. a term may have different meanings depending upon the context in which it is used. For example, “stroke” is a disorder name and that word is also present in the sentence shown in Figure 1, however, it does not mean the disorder in that sentence and hence it should not be recognized as a disorder. For these two reasons, successful NER systems recognize entities based on the contexts in which they are present in sentences and not simply by matching known entity names. In addition, for some NER tasks, like identifying gene names, a list of entity names is also never complete because new genes are being continuously discovered and given new names.

In [12] four different approaches of developing NER systems have been listed. These approaches are dictionary based, rule based, machine learning based classification model, and machine learning based sequential model. For developing the last two approaches that are classified as supervised learning methods, there is a training process that needs training data.

Some early NER systems used handcrafted rules [13-14]. For example, a rule could be “a noun phrase that follows ‘the patient has’ will be a disorder name”. However, there are numerous ways in which disorders can be mentioned in a sentence and hence it is not possible to handcraft every such rule. Moreover, such rules are not always accurate. They also require significant amount of human effort to create. For all these reasons, handcrafted rule-based approaches for NER have been largely supplanted by machine learning based approaches [15-17]. In machine learning approaches, first a training dataset is manually created by annotating several text documents with the named entities of the desired type (Figure 1 is an example of an annotated sentence). Next, a suitable machine learning method is employed that automatically learns and generalizes from this data the contexts and other characteristics based on which named entities can be accurately recognized. The trained system can then be applied to previously unseen text to recognize named entities. Such a machine learning approach is called *supervised* because it is provided manual supervision in the form of training data from which it learns. Supervised machine learning has now become the standard way of building NER systems. However, the biggest disadvantage of this approach is that it requires significant amount of manual work to annotate text in order to build the training data. For specialized text, like clinical text or biomedical literature, the annotators also need to have expertise in the subject area. Thus, supervised machine learning approach is also expensive requiring expert human labor.

On the other hand, unsupervised NER does not need any training data. These kinds of NER systems may only utilize dictionary and machine learning tools using weakly labeled-data. Thus, the implementation of these systems is less expensive than supervised systems, they need less effort to build, and they can be applied to every kind of domain including biomedical and social networks. Moreover, these kinds of NER systems can be used in different languages. These are the advantages of unsupervised NER systems, however they might not be as accurate as supervised systems.

1.2 Research Objectives and Questions

The research questions that we investigated in our research are:

- How accurately can we extract names of entities from biomedical text without requiring manual annotations?
- How does the unsupervised method to extract biomedical named entities generalize across different types of entities and different types of biomedical texts?

These are the main questions that we addressed in our research. To address the first question, we developed an unsupervised method for extracting named entities, and to address the second question we evaluated it on different types of biomedical texts and named entities.

The objectives of the research can be summarized as follows:

- Developing a novel unsupervised method for extracting biomedical named entities,
- Evaluating the developed unsupervised system with different entity classes such as biological terms (genes and proteins) and medical terms (diseases and treatments),
- Creating a tool for people to extract named entities from text of their desired domain without needing manual annotations and make it publicly available.

1.3 Research Approach

The approach presented in this dissertation leverages two large resources in order to eliminate the need for manually creating annotated training data. The first resource is Unified Medical Language System (UMLS) [18] which has more than one million clinical entity names. The second resource is a large corpus of biomedical text. If the task is to build an NER system for extracting diseases and disorders from clinical notes, our method first automatically annotates large number of clinical notes with disease and disorder entity names which are known to be unambiguous in UMLS (i.e. they do not mean anything else other than disease or disorder, say unlike “stroke” in the example of Figure 1). These form positive examples to train a machine learning method to recognize named entities. Although this process misses annotating several entities because of the variability problem pointed out earlier, however, because this is done on a very large amount of clinical text, we are able to gather sufficient number of positive examples for training. Machine learning methods also need negative examples to learn their distinction from positive examples. For this, the same process is repeated for clinical entities of *other* semantic types in UMLS (e.g., substances, anatomical structures, etc.) which form negative examples. Negative examples are selected from mentions that may belong to any semantic group other than the desired named entity class. The method works analogously if the NER task requires extracting entities of a different class.

To evaluate our method, we used three datasets representing different genres of biomedical text. The first data set used in this research, is SemEval 2014 data that includes clinical notes of four types: discharge summaries, radiology, echocardiogram, and electrocardiograph reports. Among these notes, names of diseases and disorders were to be detected. Genia and i2b2 NLP corpus are two other datasets used for testing our developed method on different types of biomedical text and named entities. GENIA corpus is a collection of Medline abstracts which is intended to represent the literature of molecular biology. It contains names of proteins, cell types, cell lines, DNAs, and RNAs. The other dataset is i2b2 NLP corpus including discharge summaries from different medical institutes in which three different classes of entities including problems (diseases), treatments, and tests are to be detected. The results presented in chapter 4 show the higher performance of our system in comparison with other unsupervised system developed in [51]. Chapter 3 describes our methods in details.

2 Background

2.1 Named Entity Extraction

According to [19], text annotation is, the practice and the result of adding a note or gloss to a text which may include highlight or underline, comments, footnotes, tags, and links.” In some scientific fields annotation is similar to meta-data, because they provide information about the text or part of a text document. In [20], the authors identified four primary applications of text annotation: facilitating reading and later writing tasks, eavesdropping on the insights of other readers, providing feedback to writers or promoting communication with collaborators, and calling attention to topics and important passages.

In recent years, the use of natural language based systems such as question-answering, summarization, and translation has grown up fast. The main concentration of these kinds of systems is analysis of natural language. Moreover, processing huge amounts of text is another requirement for them. Automatic natural language annotation is often used in these systems and to handle huge amount of data. In natural language processing, automatic annotation is a process of extracting information from text documents by machines. Named Entity Recognition (NER) [21], is a task of information extraction and is being extensively used to annotate texts. NER seeks to locate and classify elements such as names of persons, organizations, etc. among text documents. Application of NER in extracting information from biomedical text is crucial, and it prepares that information for further analysis; such as, relation extraction between biological entities. For example, by detecting names of genes and proteins we are able to study the association of gene clusters with the biological content provided by corresponding literature.

The approaches presented in our study are based on a dictionary. The dictionary that we used to extract named entities is Unified Medical Language System (UMLS). Its meta-thesaurus contains

1 million biomedical concepts and 5 million concept names. After the description of UMLS, next in this chapter, we present related work on named entity extraction.

2.2 Unified Medical Language System

The Unified Medical Language System (UMLS) is a collection of many controlled vocabularies in the biomedical sciences. UMLS provides facilities for natural language processing and a mapping structure among these vocabularies; thus, allowing one to translate among the various terminology systems. The UMLS was designed and is maintained by the US National Library of Medicine. It is updated quarterly and freely available [18].

UMLS consists of Knowledge Sources (databases) and a set of software tools including:

- **Metathesaurus**: Terms and codes from many vocabularies, including CPT, ICD-10-CM, LOINC, MeSH, RxNorm, and SNOMED CT
- **Semantic Network**: Broad categories (semantic types) and their relationships (semantic relations)
- **SPECIALIST Lexicon and Lexical Tools**: Natural language processing tools

Semantic Network and Lexical Tools has been used to produce the Metathesaurus. Metathesaurus production involves [22]:

- Processing the terms and codes using the Lexical Tools
- Grouping synonymous terms into concepts
- Categorizing concepts by semantic types from the Semantic Network
- Incorporating relationships and attributes provided by vocabularies
- Releasing the data in a common format

Mentions in the UMLS have been separated in 15 semantic groups. These groups are as follows:

1. Activities and Behaviors
2. Anatomy
3. Chemicals and Drugs
4. Concepts and Ideas
5. Devices
6. Disorders
7. Genes and Molecular Sequences
8. Geographic Areas
9. Living Beings
10. Objects
11. Occupations
12. Organizations
13. Phenomena
14. Physiology
15. Procedures

In the current research, we concentrated on names of biomedical entities.

2.3 Related Work

In this section, we first review past work in supervised methods in extracting biomedical named entities. Supervised methods need manually annotated training data. Next, unsupervised named entity recognition systems and specifically unsupervised biomedical NER from past work are reviewed. Unsupervised methods do not need any manually annotated training data.

2.3.1 Supervised Named Entity Recognition

Supervised named entity recognition methods use manually labeled training data for training machine learning systems for extracting named entities. These methods need very accurate labeled data. Supervised NER methods have been developed based on two common techniques: sequence labeling based and classification based. Conditional random fields (CRFs) [23] and Hidden Markov Models (HMMs) [24] are sequence labeling based machine learning methods that have been extensively used in NER systems. These kinds of methods work by labeling the entire sequence of words in a sentence whether they are part of named entities or not. Machine learning classifiers such as support vector machines (SVMs) [25], decision trees, etc. have been used in classification based NER systems. These types of methods work by classifying a word or a phrase in a sentence whether it is a named entity or not.

2.3.1.1 Supervised Biomedical Named Entity Recognition

In Biomedical Named Entity Recognition (BM-NER), there are two main concentrations: extracting names of genes, proteins, and relevant biological terms, and also identifying names of diseases, drugs, and other medical terms. In [51], these two NER systems have been denoted as biological NER and medical NER respectively. The NER system that were developed early were rule based or lexicon based [26-33]. One of the NER systems in this area is MedLEE [26], that is a general natural language processing based system for analysis of clinical text, and encoding and mapping terms to a controlled vocabulary. With some modification from MedLEE, Friedman et al [32] developed GENIES to extract molecular pathways from journal articles. EDGAR is another natural language processor that detects information about drugs and genes related to cancer from biomedical literature [30]. The next NER system which has been very successful is AbGene [27]. This system extracts names of genes and proteins. Another common tool for BM-NER is developed by National Library of Medicine (NLM), known as MetaMap [33]. This tool extracts mentions based on UMLS Metathesaurus. Many of these systems highly depend on lexical knowledge resources such as UMLS [34] and GO [35]. Newly concept detection and term normalization to UMLS in clinical text are provided by cTAKES [36].

However, the rule-based NER systems lack portability and robustness. Creating and maintaining rules also requires a significant amount of human time and effort and is thus also expensive. Rules need to be modified or new rules need to be created every time the data changes. Rule-based methods are often domain and language specific and cannot be adapted well to new domains [37]. Thus, supervised methods have been developed to solve these problems.

In recent years, many data driven based approaches have been developed in BM-NER with having access to annotated data. GENIA corpus [38] has accelerated related research in biological NER using different kinds of supervised learning models, such as Support Vector Machines (SVMs) [39-41], Hidden Markov Models (HMMs) [42], and Conditional Random Fields [43-48]. GENIA was used as a dataset in the shared task of Bio-NLP/NLPBA 2004 [38], and nine teams participated in that competition. Another task that participants were supposed to extract gene mentions [49] was in BioCreAtIvE challenge [50]. These kinds of tasks are being held every year with new challenges, advancing the field with relevant information extraction tasks, such as gene normalization [52] and bio-event identification [53]. Most of developed models in these challenges are supervised based on SVM and CRF.

Another domain in BM-NER is medical domain, in which publicly available corpus NER evaluation was created in the i2b2 challenge 2010 [54]. In this event, 22 teams submitted their developed models including supervised and semi-supervised ones [53]. Before the availability of i2b2 corpus, latest research was focused on evaluating, extending, and comparison with MetaMap and its predecessor MMTx [55]. Abacha and Zweigenbaum did some modifications on MetaMap and compared it with statistical based methods like CRF and SVM [56-57]. A fuzzy matcher was implemented by Patrick et al. for mapping terms to UMLS concepts [58]. Before creating i2b2, a dataset of clinical progress notes with 11 concept categories were annotated by Wang, and it was used for evaluating CRF [59]. They also provided a cascading system that combines an SVM, a Maximum Entropy, and a CRF model to reclassify extracted entities to improve accuracy of classification [60]. Most recent advances in clinical entity recognition follow the trend of supervised learning, combined with ensemble system [61], and large scale feature engineering [62-63].

Another shared task for medical NER was SemEval 2014 Task 7 in which teams were supposed to identify names of diseases and disorders. In SemEval 2014 Task 7, a total of 21 teams participated including our team. They used various approaches for tackling the tasks, ranging from purely rule-based, unsupervised [64-65], to a hybrid of rules and machine learning classifiers that acquired high accuracy using SVM and CRF [66-67]. Complete results of SemEval 2014 are presented in Chapter 3.

The main problem with supervised methods is providing training data. These kinds of data are manually annotated text used to train machine learning tools. These data must be precisely annotated by experts for training an accurate system. This process is labor intensive and increases the cost of developing such system. In contrast, unsupervised NER systems are an attractive alternative which do not require any manual annotations and can hence be developed with very little cost and manual effort.

2.3.2 Unsupervised Named Entity Recognition

In this subsection, we describe some unsupervised NER systems developed for the general domain. In the next subsection, we describe unsupervised NER systems for the biomedical domain.

Several researchers have developed unsupervised NER systems. In [68] and [69], authors developed NER methods based on heuristic rules and lexical resources like WordNet [70]. Rau [68] implemented an algorithm that extracts company names automatically from financial news. The author implemented a good algorithm by combining heuristics, exception lists, and extensive corpus analysis. In her research, she addressed two common problems which are extracting company names from text and recognizing subsequent references to a company. The algorithm generates the most likely variations that those names may go by, for use in subsequent retrieval. The implemented algorithm was tested over one million words of financial news, and extracted names of companies with 95% precision. It succeeded in extracting 25% more companies than were indexed by a human. Coates-Stephen proposed an approach that used an internal structure of names and the descriptive information that regularly accompanies them to produce lexical and knowledge base entries for unknown proper names. [68] and [69] are early works in this area.

In more recent works, the authors proposed a procedure to automatically extend ontology with domain specific knowledge [71]. Alfonesca and Manandhar stated that the main benefit of their approach is its ability to be applied to any language. They implemented an algorithm based on word-sense disambiguation procedure. The method used in their approach is Aggire's method [72]. Aggire's method consists of the following steps:

- Generating query containing words

- Submit the query to Internet search engine and collecting results
- Download documents and calculate frequencies of words
- Store list of words and frequencies

The reason that they used Aggire’s method is that WordNet does not have topic headers, and they used it to create them. After collecting headers, they weighted each word to provide contextual support, and finally to extract named entities.

Nadeau et al [73] proposed an unsupervised NER system that was made of two modules. The first one is used to create large gazetteers of entities, such as a list of cities. The second module used simple heuristics to identify and classify entities in the context of a given document (i.e., entity disambiguation). Generating gazetteers is a task of automatically generating lists of entities that has been investigated by several researchers. After running module one or generating gazetteers, resolving ambiguity was performed in module two. The list lookup strategy is the method of performing NER by scanning through a given input document, looking for terms that match a list entry. Their system was evaluated on two corpuses, the MUC-7 Enamex and Car Brands. They believed that the good performance of gazetteer generation, combined with ambiguity resolution, on an entirely new domain emphasizes their domain-independent character and shows the strength of the unsupervised approach.

Sakine and Nobata [74] developed a named entity tagger using dictionary and pattern based rules for 200 categories of named entities. The dictionaries that they used were created by accumulating 130,000 instances of each category from the Web, newspapers, and other sources, and a dictionary with 50,000 common noun phrases as well. The rules were used to identify entities which could not be matched by dictionary. The rules used patterns including four components:

- literal string
- word class
- part of speech
- current tagged NE label

The results showed that their proposed approached achieved 80% in precision and 72% in recall.

Shinyama and Sekine believed that distribution of words in news articles is a way to obtain rare named entities [75]. Based on Zipf's law [76], they found that most name entities, which are a large portion of vocabulary, are rarely used. Therefore, it is not easy for NER system developers to continue with a contemporary set of words, however a large number of documents are provided for learning. In this study, they proposed a method to strengthen the lexical knowledge for NER by using synchronicity of names. The documents used in their study that were comparable are less restricted than parallel documents, and also they are more available. A significant characteristic that named entities tend to have in comparable documents is that they were preserved across documents because it is hard to paraphrase them. Thus, in two sets of comparable documents the distribution of some specific names in one set looks like another document set. Based on this characteristic, authors concentrated on time series of distribution of words, and they hypothesized a time series of a certain word must be same as time series of the same word in another document. Then, they calculated similarity of time series of distributions among documents. Finally, the highly-ranked words were taken as NE, which was obtained by measuring similarity between time series of a certain word in different documents. They successfully achieved rare named entities with 90% accuracy.

The other category of methods, which are more recent, are in fact weakly supervised instead of unsupervised. These methods used a bootstrapping-like manner to strengthen the models. The first important study in this category was done by Collins and Singer [77]. In their study, they used seven simple seed rules for training a classifier instead of a large number of rules. The only supervision in this study is in the form of seven rules that are:

- full-string = New York -> Location
- full-string = California -> Location
- full-string = US. -> Location
- contains (Mr.) -> Person
- contains (Incorporated) -> Organization
- full-string = Microsoft -> Organization
- full-string = I.B.M. -> Organization

They used this approach to gain leverage from natural redundancy in the data: for many named-entity instances, both the spelling of the name and the context in which it appears are sufficient to determine its type. The classification approach that was used in their study was based on “Rote Learning” method [78].

A multi-level bootstrapping method was introduced by Riloff and Jones [79]. In this study, they generated semantic lexicon and extraction patterns simultaneously which usually are two requirements of dictionaries. A mutual bootstrapping technique was developed to alternately select the best extraction pattern for the category and bootstrap its extraction into the semantic lexicon, which is the basis for selecting the next extraction pattern. This method was evaluated on a collection of corporate web pages and a corpus of news articles that generated high quality dictionaries for several semantic categories. After that, several methods were developed to improve bootstrapping methods [80-82]. It must be mentioned that most of the works in this category focus only on entity classification, which assume that named entities have been identified correctly.

It is impressive that in many ways, previous studies were the beginning of unsupervised NER using word sense disambiguation, especially in classification of detected mentions. For instance, the method of bootstrapping proposed in [75] was used at first by Yarowsky [83] for word sense disambiguation. Yarovsky proposed an unsupervised learning for sense disambiguation, that when trained on un-annotated English text, competes with the performance of supervised methods that need manually annotated data. Then again, the idea of entity classification based on their context signatures [71] is similar to distributional techniques in word sense disambiguation [84], in which context of NE were used to detect word senses.

2.3.2.1 Unsupervised Biomedical Named Entity Recognition

The NER systems that were used are typically rule based or lexicon based [68, 38-43]. The only work that we are aware of in the biomedical domain is by Shaodian Zhang and Noemi Elhadad [51], who proposed a new unsupervised approach to extract mentions of diseases from clinical notes. The authors proposed a step-wise approach that does not rely on hand-built rules or examples of annotated entities, so it can be adapted to different semantic categories and text genres easily. Leveraging entity recognition terminologies, shallow syntactic knowledge, and corpus statistics were used instead of supervision. Results of applying this method on i2b2 and GENIA corpora data sets show their proposed method achieved F-scores 24.1% and 15.2% in i2b2 and GENIA test datasets respectively. Their proposed algorithm includes three steps: Seed Term Collection, Boundary Detection, and Entity Classification.

In this approach, the first step is to collect seed terms according to representation of entity classes. The seed term sets were collected from external terminologies, and were defined by user choice of semantic type/group of UMLS or specific concepts that describe the semantic domains of the classes. The second step, Boundary Detection, involves collecting candidates for entity classification. In the solution authors hypothesized entities should be noun phrases, and to remove those that are not real noun phrases, Inverse Frequent Diversity (IDF) was employed. Entity Classification is the third step of this approach. It was based on the intuition that entities of the same class tend to have similar vocabulary and context, and based on that, classification was done [51]. In [85], Elhadad used the same approach in [51], and developed a method to generate a lexicon representative of the language of members in given community with respect to specific semantic types.

In [86], another method was developed to identify semantic terms from PubMed. However, their goal was to extend existing terminologies, a task which is different from NER. The first step in their method is to obtain headwords uniquely corresponding to concepts. The concept of a phrase is mostly determined by the headword. Thus, the procedure guarantees that the same concept phrases are investigated. Extracting candidate terms is the next step, which is done by using linguistic patterns. This process also removes the headword, or it tries to find neighboring terms which are semantically linked to the headword. However, the terms extracted from the linguistic patterns may be noisy; thus, an SVM classifier is applied to eliminate irrelevant terms in the last step [86].

The advantage of our method over these methods in biomedical named entity extraction is that we used a machine learning based tool, in which we generated weakly-labeled data for the training the machine learning tool. This allowed us to develop more accurate systems than those that were developed before. The results shown in chapter 4, compared the performance of these systems.

2.4 Our Previous Work

In our previous work, we developed a supervised NER system that utilized the machine learning method of CRFs. We developed a BM-NER for participating in SemEval 2014 Task 7 [4], Natural Language Processing competition. In that competition, we were supposed to extract names of diseases and disorder from clinical notes.

In that study, we used data sets provided by SemEval 2014 organizers. SemEval is an ongoing series of evaluations of computational semantic analysis systems. In the competition, we participated [46] in Task 7 "Analysis of Clinical Text" [87] in which participants were supposed to extract names of disease and disorders by developing supervised, un-supervised, or semi-supervised models. Based on SemEval rules diseases and disorders are scattered among 12 semantic types in "Disorder" semantic group shown in Table 1.

Table 1 Semantic types of disease/disorders accepted by SemEval

Semantic Type	Code
Acquired Abnormality	T020

Anatomical Abnormality	T190
Cell or Molecular Dysfunction	T049
Congenital Abnormality	T019
Disease or Syndrome	T047
Experimental Model of Disease	T050
Injury or Poisoning	T037
Mental or Behavioral Dysfunction	T048
Neoplastic Process	T191
Pathologic Function	T046
Sign or Symptom	T184

The data provided by SemEval includes two sets. The first one is training data set that contains 199 clinical reports. The other data set is test set that have 133 clinical reports. Training data consists clinical reports of four types: discharge summaries, echocardiogram, electrocardiogram, and radiology reports, while test data set contains only discharge summaries. The distribution of reports in training data set is shown in Table 2. These datasets were manually annotated with diseases and disorders. In the current work, we used these gold-standard annotations only for evaluation and not for training.

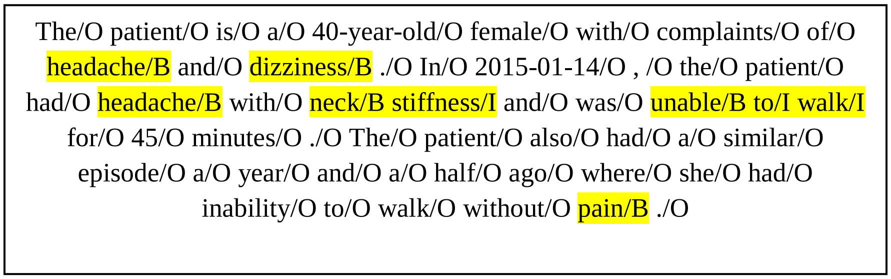
Table 2 Distribution of reports in training data set

Type of Report	**Count (%)**
Discharge Summary	61 (30.7%)
Echocardiogram	54 (27.1%)

Electrocardiograph	42 (21.1%)
Radiology	42(21.1%)

CRFs are kinds of undirected probabilistic graphical models. A CRF is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence. The main advantage of CRFs against HMMs is their conditional natures that result in relaxation of the independence assumptions required by HMMs in order to ensure tractable inference [88]. They can be used to model relationships between observations and make a robust model to recognize the sequence relationships. Most of the times CRFs have been used to predict labels or to parse sequence of data points. One of the most important applications of CRFs is in natural language processing, gene prediction, and image processing. In natural language processing CRFs usage is growing fast, and they have been used for shallow parsing and named entity recognition [88]. They can incorporate a large set of arbitrary and non-independent features while still having efficient procedures for non-greedy finite-state inference and training. CRFs have been indicated robust and reliable in different sequence modeling tasks including named entity recognition [89].

Also in our research, we used *BIO* approach to label words through sentences. The goal is to predict labels (*B*= beginning, *I*= inside, or *O*= outside) of each word in text. In Figure 2, a paragraph from a discharge summary has been tagged with related *BIO* labels. Words tagged as *B*s and *I*s are desired named entities. A desired named entity starts with tag *B* and continues with tag *I*, for mentions with more than one word. Single word named entities tagged with *B*, and other words that are not part of desired mentions tagged with *O*.



The/O patient/O is/O a/O 40-year-old/O female/O with/O complaints/O of/O headache/B and/O dizziness/B ./O In/O 2015-01-14/O , /O the/O patient/O had/O headache/B with/O neck/B stiffness/I and/O was/O unable/B to/I walk/I for/O 45/O minutes/O ./O The/O patient/O also/O had/O a/O similar/O episode/O a/O year/O and/O a/O half/O ago/O where/O she/O had/O inability/O to/O walk/O without/O pain/B ./O

Figure 2 Example of tagging a paragraph in our system

To train the CRF, we used a set of features listed in Table 3.

Table 3 Features used for training CRF

ID	Feature Name	Tool used to extract
1	Word	Programming Language
2	Next word	Programming Language
3	Previous word	Programming Language
4	POS tag of Word	Stanford NLP
5	POS tag of next word	Stanford NLP
6	POS tag of previous word	Stanford NLP
7	Next two words	Programming Language
8	Previous two words	Programming Language
9	Length of the word	Programming Language
10	Semantic group of the word	UMLS
11	Semantic group of next word	UMLS
12	Semantic group of previous word	UMLS
13	Exact match of bigram	UMLS
14	Exact match of trigram	UMLS
15	Exact match of reverse bigram	UMLS

16	CUI of the word	UMLS
17	MetaMap match of the word	MetaMap
18	MetaMap match of next word	MetaMap
19	MetaMap match of previous word	MetaMap
20	Lemmatized version of the word	Stanford NLP
21	Parent of the word in dependency tree	Stanford NLP
22	Abbreviation full name	List of Abbreviations
23	Abbreviation full name exact match into UMLS	UMLS
24	Abbreviation full name semantic group	UMLS

To evaluate our developed method, we used SemEval 2014 evaluation tool, which is thoroughly explained in Section 4.2.

Our developed model [46] had achieved F-score 75.5% (precision=78.7%, recall= 72.6%). The results are shown in details in Table 4.

Table 4 Results of a supervised model

	Precision	**Recall**	**F-score**
Strict	0.787	0.726	0.755
Relaxed	0.911	0.856	0.883

At the end of the competition, our developed method came third among 24 teams around the world. Table 5 shows the results and ranking of participated teams in SemEval 2014 competition

(Our team was UWM that is highlighted). The best team, UTH_CCB, from University of Texas, Health Science Center, used more resources of annotated text to reach the best system. The method that they developed, also utilized a CRF based model for extracting names of diseases and disorders.

Table 5 SemEval 2014, Task 7 results of teams

Team Name	Run	Precision	Recall	F-Score
UTH_CCB	0	0.843	0.786	0.813
UTU	1	0.765	0.767	0.766
UWM	**0**	**0.787**	**0.726**	**0.755**
IxaMed	1	0.681	0.786	0.73
RelAgent	0	0.741	0.701	0.72
ezDI	1	0.761	0.681	0.719
ULisboa	0	0.753	0.663	0.705
BioinformaticsUA	0	0.813	0.605	0.694
ThinkMiners	0	0.734	0.65	0.689
ECNU	0	0.754	0.611	0.675
UniPI	2	0.712	0.601	0.652
UNT	0	0.647	0.628	0.638
CogComp	1	0.639	0.529	0.579
TMU	0	0.524	0.576	0.549
MindLab-UNAL	2	0.561	0.534	0.547
IITP	0	0.5	0.479	0.489
SZTE-NLP	1	0.547	0.252	0.345
QUT_AEHRC	0	0.387	0.298	0.337
KUL	0	0.655	0.178	0.28
UG	0	0.114	0.234	0.153

3 Methods

3.1 Overview

As mentioned before, the purpose of this research is to reduce human effort of annotation for building entity extraction systems for biomedical domains. Our previous work showed a supervised model that reached considerably high accuracy in comparison with other developed methods by teams participated in SemEval 2014 NLP competition. In this chapter, we present our proposed methods for detecting named entities without any training data that is manually annotated. Figure 3 shows the overall approach to our proposed method for unsupervised biomedical named entity recognition. In our approach, the first step is to extract entities, which is "entity detector". We create weakly-labeled data using "Unambiguous exact matching" algorithm based on UMLS. Then we trained a machine learning tool using weakly-labeled data. After this step, "Entity trigger" algorithm used to send terms from corpus to the classifier to determine whether they belong to the desired entity class or not. Next, new entities are added to the weakly-labeled data and the machine learning tool is further trained iteratively. Extracted entities, outcomes of "Entity Detection", are sent to the "Boundary Expansion" to find correct boundaries of each extracted mention.

We developed two types of boundary expansion methods: WordNet based and Machine learning based methods. In the first one, we used an algorithm to find right boundaries of mentions using WordNet and UMLS, while in the latter we used a machine learning tool to correct boundaries. This machine learning tool was trained using UMLS terms. These approaches are described in detail in the next sections. Figures 3-5 illustrate our developed approach.

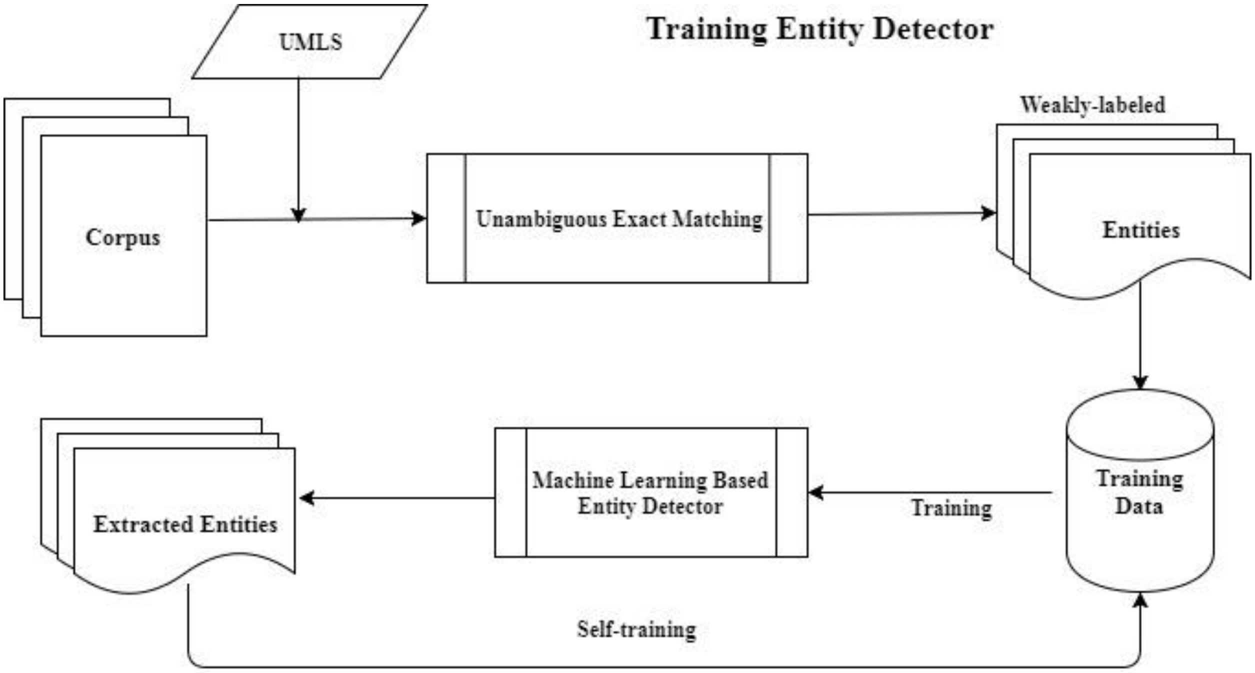
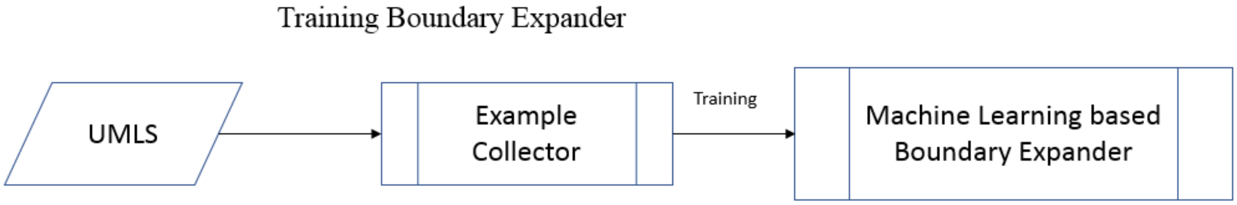


Figure 3 Overall approach for training named entity detector



Training Boundary Expander
UMLS
Example Collector
Training
Machine Learning based Boundary Expander

Figure 4 Overall approach for training boundary expander

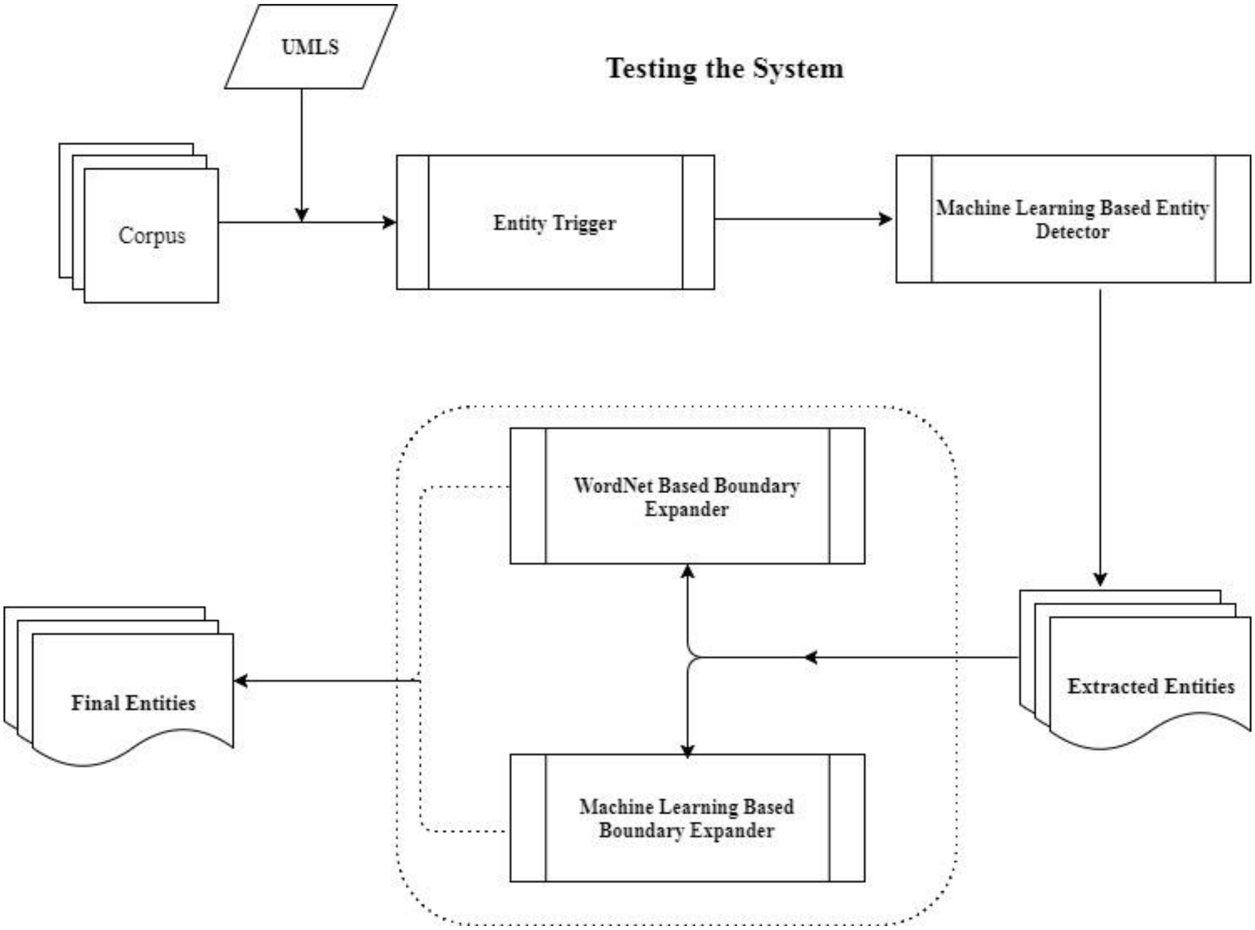

UMLS
Testing the System
Corpus
Entity Trigger
Machine Learning Based Entity Detector
WordNet Based Boundary Expander
Final Entities
Extracted Entities
Machine Learning Based Boundary Expander

Figure 5 Overall approach for testing the unsupervised NER system

3.2 Entity Detection

In this section, we describe our machine learning based tools for entity detection. The first step is to generate some training data with high precision. Next is to use that data for training the tools, and finally we developed methods for detecting correct boundaries of mentions.

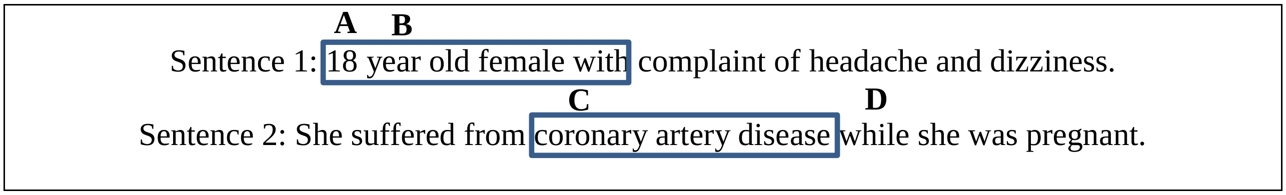
3.2.1 Creating Weakly-Labeled Training Examples

Firstly, we used a dictionary based method to generate weakly-labeled annotated clinical notes. The dictionary used in our research is Unified Medical Language System or UMLS. The next section describes the algorithm of generating weakly-labeled training examples.

3.2.1.1 Unambiguous Exact Matching

The algorithm of generating weakly-labeled training examples is called "Unambiguous Exact Matching". Here "unambiguous" refers to those mentions in dictionary which belong to only and only one semantic group such as "Disorders". This concept is discussed later in this section.

Our suggested method basically is a procedure of mapping all words (or unigrams) and terms (or n-grams, *n>1*) found in the text to a dictionary. This process begins with determining a window on text and moving it through all sentences in the documents. A window is a series of words that begins with word at position *1* and ends with word at positions *n* based on n-gram size. Maximum size of window is *n* that determines number of words of terms to be mapped to UMLS meta-thesaurus. After locating window, the algorithm maps all words inside the window to UMLS. If there is no match for that term, size of window is subtracted by one. Now the number of words inside the window is *n-1*. This process goes on until the size of window becomes one. In other words, the process stops when there is only one word left. To find matches of single words only nouns, detected by their part of speech (POS) tags were considered to be mapped to the UMLS, because we hypothesized desired mentions are nouns. If the algorithm could not find any matched term in the dictionary for the window, then the starting point of window increases by one, and the process repeats for the new window. If there is any matched term, the new window begins with the word right after the matched term. In the following example, exact matching procedure has been shown.



Sentence 1: 18 year old female with complaint of headache and dizziness.

Sentence 2: She suffered from coronary artery disease while she was pregnant.

Figure 6 Exact Matching Method

In Figure 6, sentence 1, the series of words in the window starts at point *A*. At first there are $n=5$ words in the window. The process starts with mapping that and ends when window size is one and there is no match for that single word. Then it moves the cursor to the next position that is point *B*. In the second sentence algorithm finds a match for the n-gram inside the window. After finding a match the cursor moves to the next point right after the matched mention that is point *D*. By using this method through all text documents, we were able to generate weakly-labeled clinical notes.

Another important thing that must be mentioned is that we put a restriction on matched terms. In the UMLS terms might have more than one semantic type. That means a mention can belong to more than one semantic groups. For example, “distress” is a “disease” and a “physiological process”. Because of that, we restrict matched terms to be of only one semantic type. Thus if “distress” word is found in the text, it will not be matched to any UMLS term in our method, because it is known to be ambiguous, so the algorithm is called “unambiguous exact matching”. Because of this restriction, we can be sure that all the matched terms are of the desired semantic type only and do not mean something else in the text. This is a distinct feature of our method, in the previous work [51] had simply included all the mentions, including ambiguous ones, which leads to incorrect training examples. In our method, we generated positive examples from those mentions that belong to only and only one semantic group. On the other hand, negative examples are those mentions that belong to all other semantic types except the desired one that named entities belong to.

3.2.2 Training a machine learning tool

After running the exact matching algorithm, we have training data that we call weakly-labeled that can be used to train a machine learning tool for detecting mentions Using a machine learning tool for entity detection can be done by using classification based models or sequence labeling based models. In this framework, we used classification based model, because many positive examples may be missed. The developed methods are presented in the next sections.

3.2.2.1 Sequence Labeling Method vs. Classification

The first step after generating weakly-labeled training data, is to develop a machine learning tool for detecting mentions that were missed by the exact matching algorithm.

The most important observation is that because we considered mentions with only one semantic type, the system missed many mentions that are terms to be extracted. Sequence labeling methods expect every word of a sentence to have a correct label in the training data. Thus, the words which are not labeled as named entities will be considered as negative examples. Given that our process of automatically creating training data misses some entities, the missed entities will then go in with incorrect labels if a sequence labeling method is used to train, thus adversely affecting the training process. Hence in the proposed project, instead of using a sequence labeling method such as CRF, we used a classification method such as Decision Tree and Support Vector Machine (SVM) [90] [i]. In a classification method, the missed entities do not automatically become negative examples.

Furthermore, many of detected mentions by exact matching are those that have one or more missing parts. These mentions may be partially matched to vocabulary. For example, there is no exact match for “gas discomfort” in UMLS, in “disease/disorder” entity class, but there is one for "gas". In that case the exact matching fails to detect "gas discomfort" as a disease although it could have detected "gas" as a disease.

By analyzing results of exact matching and weakly-labeled models, we realized that there were many typos and different derivatives of words for related diseases, and moreover, around 10% of mentions were discontinuous. For instance, "left atrium is moderately dilated" contains disease name that its preferred term in the UMLS is "left atrial dilation". In this case, not only a new method is needed to detect "dilated" as "dilation", but also an algorithm is required to handle discontinuous mentions such as "left atrium dilated" in "left atrium is moderately dilated".

3.2.2.2 Learning Classification Methods

To avoid distracting machine learning tool by imperfect training examples (weakly-labeled), we developed a novel method based on learning classification tools for entity detection, which is the most significant part of our research, and an algorithm to handle discontinuous mentions. The proposed approach constitutes three components: a learning classification tool, border detection, and detecting discontinuous mentions. These methods are described in the next sub sections.

Classification learning methods require examples to be represented in terms of features from which they can learn statistical regularities and patterns to distinguish between positive and negative examples. We used Decision Tree as our main classification method, because we found out that by experimenting different classifiers such as SVM and decision trees and got better results. We used the Weka software [91] [ii] to implement "Random-Subspace" decision tree which is publicly available for free. The decision tree classifier trained using the positive and negative examples generated in "Unambiguous Exact Matching" were then applied to classify each term in new text whether it is a named entity of the desired type or not. Hence, positive examples are those detected mentions corresponding to only one semantic group in the UMLS. In UMLS, there are numerous mentions that are linked to multiple semantic groups. We applied the classifier to only noun phrases because named entities are almost always noun phrases.

Detecting mentions of the desired named entity type which in the dictionary are ambiguous (terms belong to multiple semantic groups) is the main purpose of this classifier (unambiguous terms can be detected easily as in our previous step). Terms with unique semantic types, generated by Exact Matching, and such ambiguous terms will be found in text in similar contexts. The machine learning method will learn these contexts from the weakly-labeled training data of unambiguous terms and then will be able to detect terms which are known to be ambiguous. The contexts are provided to the machine learning method in terms of features.

After some experiments, we found out that SVM based classifiers work better with large number of features. These kinds of cases could be found in classifying sentences that may contain words and other n-grams among all documents as features. Also after doing some trials on data and checking different classifiers, such as Bayesian methods, the most accurate results were acquired by Decision Tree based methods. These kinds of classifiers use decision trees to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees [92].

The type of decision tree that was used as classifier in our research is “Random-Subspace.” In machine learning the random subspace method, also called attribute bagging or feature bagging, is an ensemble learning method that attempts to reduce the correlation between estimators in an ensemble by training them on random samples of features instead of the entire feature set. In ensemble learning one tries to combine the models produced by several learners into an ensemble that performs better than the original learners. One way of combining learners is bootstrap aggregating or bagging, which shows each learner a randomly sampled subset of the training points so that the learners will produce different models that can be sensibly averaged. In bagging, one samples training points with replacement from the full training set [93]. The random subspace method is similar to bagging except that the features ("attributes", "predictors", "independent variables") are randomly sampled, with replacement, for each learner. Informally, this causes individual learners to not over-focus on features that appear highly predictive/descriptive in the training set, but fail to be as predictive for points outside that set. The random subspace method has been used for decision trees; when combined with "ordinary" bagging of decision trees, the resulting models are called random forests [93-94].

We used 55 features in this research which include surrounding words, words of the mentions, part of speech tags, lemmatized and stemming forms, and sematic types of words. All features used for training the classifier are listed in Table 6. By using the classifier, we could reach a significant improvement in the results that are shown in the next chapter.

Table 6 List of features used in classifier

Number of features	Feature Name	Tool used to extract
5	Mention (includes five words or less)	Programming Language

3	Next three words	Programming Language
3	Previous three words	Programming Language
5	POS tags of Words of Mention	Stanford NLP
3	POS tag of next words	Stanford NLP
3	POS tag of previous words	Stanford NLP
5	Semantic types of the words	UMLS
3	Semantic types of next words	UMLS
3	Semantic types of previous words	UMLS
5	Lemmatized of words of mention	Stanford NLP
3	Lemmatized of next three words	Stanford NLP
3	Lemmatized of previous three words	Stanford NLP
5	Stemmed of words of mention	Programming Language
3	Stemmed of next three words	Stanford NLP
3	Stemmed of previous three words	Stanford NLP
Total: 55		

3.2.3 Self-training

In order to further improve the performance of our system, we applied the trained classifier on the training corpus itself to gather more positive and negative examples which are in turn used to train the classifier again. This way of training is also known as self-training and has been shown to be helpful in improving performance. The process starts firstly by training the classifier with the weakly-labeled training data. After training and applying it to clinical notes, we were able to detect ambiguous mentions and to check whether they are named entities of the desired type or not. Then newly extracted terms (from ambiguous mentions) were added to the old training data set. By using the new training data set and repeating the same procedure, new mentions were extracted, and the performance was improved. The iterative process is repeated a few times until no new examples are gathered.

3.2.4 Testing the System

To apply our system, potential entities are first collected by an "Entity Trigger". The terms in the text that match terms of the required semantic types in UMLS, including partially, are triggered as possible entities. These are then sent to the trained classifier for entity detection (described in the last section). The classifier generates a number between 0 and 1. If the term could pass the threshold, then it can be categorized as a desired mention. The threshold for each entity class was determined by experimenting different numbers to get the best performance.

3.3 Boundary Expansion

Many named entities detected by the entity detector may have missing parts or incorrect boundaries. Since, many named entities may have missing parts or incorrect boundaries. For example, "effusion" was detected by classifier as a named entity in diseases class, but the mention that has correct boundaries is "pericardial effusion". To solve this problem, we developed two methods for detecting correct boundaries of detected mentions. The first method utilized WordNet and Elasticsearch, and the second one used a classifier to detect correct boundaries of mentions trained by examples generated from UMLS mentions. The next sections describe the two methods.

3.3.1 WordNet Based Method for Boundary Expansion

For developing our boundary expansion method, firstly we utilized WordNet [70] database to find derivatives of words. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synsets or their members [95]. Words derivatives vary in sentences based upon their part of speech tags and positions, hence mentions might be different from their relevant term in UMLS. For example, in "Living beings" semantic group in UMLS, "mice" does not exist but there is "mouse" instead, or "dilated left atrium" is in UMLS as "left atrial dilation". By using WordNet, lemmatization, and stemming techniques we extracted all possible derivatives of words and their combinations of UMLS mentions. Example 1 shows how WordNet was used to obtain all derivatives of words and their combinations in a mention.

Using WordNet to extract derivatives of words: left, atrial, and dilation.

- WordNet derivatives of each word:
    - Left -> Left
    - Atrial -> Atrial, Atrium
    - Dilation -> Dilation, Dilate, Dilatation

All Combinations:

- Left atrial dilation
- Left atrial dilate
- Left atrial dilatation
- Left atrium dilation
- Left atrium dilate
- Left atrium dilatation

Example 1 Using WordNet to extract derivatives, and making new combinations

In this research, we used Elasticsearch [96] to get access to mentions as fast as possible, and WordNet was used for extracting different forms of mention. The method was not only used for correcting boundaries, but also it was used to find discontinuous mentions that are around 10% of all terms (described in Section 3.4). Our method starts with indexing mention in some Elasticseach indexes. Then it uses some similarity functions to map mentions to UMLS terms.

Before mapping mentions to UMLS, we used Elasticsearch to store and to access them quickly. Elasticsearch is a search engine based on Lucene. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. The main purpose of using Elasticsearch is to store tremendous amount of data and to access them as quick as possible. It can save millions of documents (mentions) and can access them very quickly [97].

In order to access stored data in Elasticsearch, for each record a “key” must be defined. A key plays an important role in our approach, because selecting an appropriate key may lead us to get more proper outcomes. Choosing inappropriate keys distracts the whole process and may result in redundant outcomes, consequently significant time consumption of process.

To select a proper key for each mention in UMLS, we relied on facts of results generated by Exact Matching and the classifier. Results contain two types of scores: Relaxed and Strict. Exact scores indicate accuracy of boundaries of mentions from both left and right sides while inexact scores include all mentions correctly detected not only from both sides but also from the left or right. Based on that fact, we decided to use the most-left and the most-right words in mentions to be keys in Elasticsearch indexes (at least one side, left or right, is correct). Therefore, each mention has been saved into an Elasticsearch index twice and is accessible by its most-left or right words. A simple procedure of storing a mention in Elasticsearch is shown in Example 2.

Mention: Left atrial dilation

- Keys:
  - Most-left word: left
  - Most-right words: dilation, dilate, dilatation

Example 2 Selecting key of a mention

In Example 2, there are totally four keys for mention "left atrial dilation". The most-left word of the mention is "left" that have one synonym in WordNet as "left", and word dilation has three different derivatives extracted by WordNet that shows another three different keys.

Table 7 All variations of mention "left atrial dilation" to be saved in Elasticsearch index

Key	**Mention**	**Key**	**Mention**
Left	Left atrial dilation	Dilation	Left atrial dilation
Left	Left atrial dilate	Dilation	Left atrium dilation
Left	Left atrial dilatation	Dilate	Left atrial dilate
Left	Left atrium dilation	Dilate	Left atrium dilate
Left	Left atrium dilate	Dilatation	Left atrial dilatation
Left	Left atrium dilatation	Dilatation	Left atrium dilatation

Table 7 shows a sample mention and its variations to be saved in Elasticsearch index.

In this approach, there may be many terms with same key. This ability not only is not an issue, but also it increases flexibility. High flexibility allows us to reach all mentions relevant to the keys of a mention being considered. For faster retrieval, indexes were separated based on the number of words in mentions (n-grams). This avoids extracting redundant mentions from the index. Mentions with two words or bigrams, trigrams, four, five, and six grams were saved in separate indexes. Moreover, all stop words such as determiners, to be verbs like am is, are, and adverbs must be removed from the mentions before storing in the indexes. By using the proposed strategy, the procedure spends less time for processing. In the next section, we have described how indexes were used for boundary expansion.

Our developed boundary expansion method in this research extracts relevant cases out of Elasticsearch indexes based on keys and number of words in each mention.

Correcting boundaries starts with extracting all noun phrases around the detected mentions. These noun phrases must be selected from only one side of the mentions for each process, because all mentions have correct boundaries from at least one side. If the mention expands to the right, the key to Elasticsearch index will be the most-left word and vice versa. The method continues by processing the biggest noun phrase until reaching the initial term. Example 3 shows how noun phrases of mention are formed.

Mention: thickened.

Sentence: Left arterial ventricular is thickened

UMLS term: Arterial ventricular thickened.

Example 3 Forming of noun phrases

As indicated in Example 3, suppose the mention “thickened” is detected as an entity using the method described in Section 3.2. The noun phrases around it are “left arterial ventricular thickened", "arterial ventricular thickened", and "ventricular thickened". After extracting all noun phrases and removing all stop words (determiners, adverbs, to be verbs, and symbols), the process continues with mapping the generated noun phrases to the mentions stored in indexes based on n-grams. If there is not any match for the biggest noun phrase, the process proceeds to match the next noun phrase until it gets a match or the initial mention.

Flexibility of mapping mentions with different number of words is another benefit of the proposed method. For instance, it is possible to map trigram to all other n-grams stored in Elasticsearch. This ability allows the developed method to find corresponding term of mentions in UMLS in which some words may not exist. For instance, mention of "pulmonary hypertension" in clinical notes has a corresponding term in UMLS as "pulmonary arterial hypertension".

To select the most similar/relevant mention from UMLS to detected mentions, we have assigned a similarity score to each mapping based on two factors: existence of words, and locations of them in noun phrases. To find a similarity score, we used a function mentioned in (1).

$$Sim(W_A, W_B) = \frac{L_{w_A} - L_{w_B}}{\sqrt{1 + (L_{w_A} - L_{w_B})^2}} \; when \; L_{w_A} - L_{w_B} \neq 0 \qquad (1)$$

Equation 2, calculates a similarity score for words in vector W_A $(w_i, 0 < i < n)$ in mention *A* and vector W_B $(w_i, 0 < i < m)$ in mention *B*, where *n* and *m* are number of words in mentions *A* and *B*, $L_{w_A}$is location of word in mention *A*, and $L_{w_B}$is location of word in mention *B*. the similarity score is 1 if $L_{w_A} - L_{w_B} = 0$. After calculating scores of all words in two mentions, we used equation 3 to find similarity measure of two mentions.

$$Sim(A, B) = \left(\frac{\sum_{w_i \neq w_j} p(w_i, w_j)}{n * m - 1} + \frac{\sum_{w_i = w_j} p(w_i, w_j)}{n} \right) / 2 \qquad (2)$$

Expanding to the left makes the most-right word as a key to Elasticsearch, and most-left word is used when noun phrase expands to the right. In Example 3, mention “thickened” appeared at the end of the sentence, since it expands to the left. Then mapping starts with the noun phrases with bigger number of words, and it continues until a match found in Elasticsearch. This example is thoroughly explained in the next subsection.

After extracting terms from Elasticsearch index, similarity probabilities must be calculated for each noun phrase. If similarity function exceeds a threshold, then the noun phrases will be replaced by the initial mention. In Table 3, similarity thresholds are shown that were set based pilot studies. We some numbers between 0 and 1, and finally we achieved the most accurate results using thresholds listed in Table 8.

Table 8 Threshold of WordNet based algorithm

n-gram	***n*>3**	***n* = 3**	***n* = 2**
Threshold	0.8	0.85	0.95

The WordNet based method, could improve results, although it still needs to map terms to UMLS. That means the algorithm misses those words that are parts of named entities but not parts of terms in UMLS. Therefore, we developed a machine learning based method not only to capture preferred terms in UMLS, but also to find words that are not parts of terms of UML. This method is explained in the next section.

3.3.2 Machine Learning Based Boundary Expansion

Another approach that we developed for detecting correct boundaries of mentions is based on a machine learning tool. This tool uses weakly-labeled data generated from UMLS mentions.

In this approach, positive examples, were created as follows:

- If a named entity exists in another entity mention, these two make a positive example. For example, in “disease/disorder” entity class, there is disease mention “pain”. Another mention is “abdominal pain”. Thus, these two make a positive example for the machine learning tool.
- If a mention exists in another mention that is not a named entity of the desired type, then a negative example is created. For instance, “pain” is a disease, but “pain relief” belongs to “preventive procedures” semantic type in UMLS hence the pair forms a negative example.

Using these examples, the machine learning method will learn what type of words expand a named entity of the desired type to the same type which then can be used in boundary expansion.

Generating these kinds of positive and negative examples allows us to detect not only correct boundaries of mentions, but also it finds those mentions that are not desired to be extracted. Like disease mention “pain” in “pain relief”. Hence, this method helps improve the accuracy by correcting boundaries and removing extra detected mentions.

3.3.2.1 Training the Classifier

In our proposed method, we used a classifier for correcting boundaries. The classifier was trained by the generated examples extracted from UMLS. Large number of these kinds of examples are enough to train a classifier with high accuracy.

Table 9 List of features used for training Boundary Expansion tool

Number of features	Feature Name	Tool used to extract
5	Mention (includes five words or less)	Programming Language
5	POS tags of Words of Mention	Stanford NLP
5	Semantic types of the words	UMLS
5	Lemmatized of words of mention	Stanford NLP
5	Stemmed of words of mention	Programming Language
Total: 25		

Features used in this classifier are words of mentions, their semantic types, their POS tags, lemmatizations of them, and stemmed forms of those words listed in Table 9. In this research, we assumed that each named entity has maximum number of five words. If a named entity has for example three words, there must be nulls instead of the others. The classifier that was used in this approach is also a decision tree. This classifier, gets the detected mention and a noun phrase including the mention. Theses noun phrases contain words around the mention, and by using the classifier, we were able to detect the most appropriate noun phrase. Classifier gives a score number between 0 and 1, and we select the noun phrase with the highest score and higher than the threshold among all the noun phrases. By using that method, we could detect mentions that are not presented in UMLS but they are desired to be extracted. For example, in UMLS, there is mention “gunshot wound of abdomen”, but mention “knife wound of abdomen” is not presented in UMLS, and our system would otherwise only extract “wound of abdomen”. Thus, by using that method, we were able to find correct boundaries of the mention which starts at “knife” and ends at “abdomen”. Results, presented in the next chapter, show improvement of the accuracy after using our machine learning based border expansion method.

3.4 Extracting Discontinuous Entities

One of the pros of our “WordNet based” method is flexibility of mapping different n-grams to each other. In order to find discontinuous mentions, we used same approach of border expansion based on WordNet. The only things that may change are the thresholds. The reason is that there might be one or more extra words between parts mentions in the text (while they are connected to each other in UMLS). Because of that, by mapping the whole noun phrase, we may get lower similarity scores. Hence, we decrease the threshold. To use lower thresholds, there must be some extra words between parts of a mention, otherwise it is considered to be non-discontinuous. For example, in sentence “The mitral valve leaflets are mildly thickened”, “thickened” has been detected as a disease, in “disease/disorder” entity class, and because it is at the end of the sentence noun phrases selection expands to the left, consequently the most-right word must be selected as key to Elasticsearch index. The noun phrases are:

- mildly thickened
- leaflets are mildly thickened
- valve leaflets are mildly thickened
- mitral valve leaflets are mildly thickened.
- the mitral valve leaflets are mildly thickened.

After extracting noun phrases and terms from Elasticsearch regarding number of words they have, similarity score must be calculated for each mention that comes out from the index and each noun phrase. The process starts with removing stop words then follows with mapping the noun phrase with greater number of words.

By removing stop words, left noun phrases for mapping process are:

- leaflets thickened
- valve leaflets thickened
- mitral valve leaflets thickened

All listed noun phrases, would map to corresponding Elasticsearch index (bigram, trigram, and four gram indexes). If there is a match for any of them, the process stops. If not, we decrease the threshold and assume there might be a discontinuous mention. Since, we restart the process by mapping mentions to indexes with lower n-grams. The noun phrases would be “mitral valve leaflets thickened” and “valve leaflets thickened”. Noun phrase “leaflets thickened” no longer maps, because it is a bigram and we do not need to map that to unigrams (it is already done by exact matching).

Finally, “mitral valve leaflets thickened” mapped to mention “thickened mitral leaflet” in UMLS by calculating similarity score using (1) and (2).

4 Results and Discussion

In this chapter, we present results of our developed method. Furthermore, we compare our results with another unsupervised biomedical named entity recognition method developed in [51] and baseline results generated by Exact Matching. This chapter has three parts - Data Sets, Evaluation Measures, and Results.

4.1 Data Sets

The first data set used in our research is SemEval 2014. As mentioned in Chapter 2, SemEval 2014 corpus includes two sets. The first one is training data set that contains 199 clinical reports. The other data set is test set that have 133 clinical reports. Training data consists clinical reports of four types: discharge summaries, echocardiogram, electrocardiogram, and radiology reports, while test data set contains only discharge summaries. The distribution of reports in training data set is shown in Table 10. These datasets were manually annotated with diseases and disorders. In the current work, we used these gold-standard annotations only for evaluation and not for training.

Table 10 Distribution of reports in SemEval 2014 training data set

Type of Report	Count (%)
Discharge Summary	61 (30.7%)
Echocardiogram	54 (27.1%)
Electrocardiograph	42 (21.1%)
Radiology	42(21.1%)

In addition to SemEval data set, described in section 2.4, we used two other data sets to evaluate our method for its generalizability. These data sets were also used in an unsupervised biomedical named entity recognition system published in [51] which allows us to compare our results with their results in this dissertation.

The first data set other than SemEval is based on the GENIA corpus [38]. The GENIA corpus is the primary collection of biomedical literature compiled and annotated within the scope of the GENIA project. The corpus was created to support the development and evaluation of information extraction and text mining systems for the domain of molecular biology. The corpus contains Medline abstracts, selected using a PubMed query for the three MeSH terms "human", "blood cells", and "transcription factors". The corpus has been annotated with various levels of linguistic and semantic information. Biomedical entities that were to be detected are names of Proteins, DNAs, RNAs, Cell Types, and Cell Lines.

The other data set is i2b2 corpus that was created for the i2b2/VA 2010 challenge [54]. The dataset includes discharge summaries from Partners Health Care, Beth Israel Deaconess Medical Center, and University of Pittsburgh Medical Center (denoted in this paper as Partners, Beth, and Pittsburgh for short). Pittsburgh notes were used as test set in i2b2 challenge and the other two sources as training set. All records in the dataset have been fully de-identified and manually annotated for concept, assertion, and relation information. In this paper, only concept annotations are used with three categories of entity annotations: Problem, Treatment and Test. In the next section, we present our evaluation methods. In Table 11, more details about i2b2 and GENIA datasets are presented.

Table 11 Data presented in i2b2 and GENIA corpora

Corpus	No. of Documents	No. of Sentences
i2b2 (Pittsburgh)	477	27,627
i2b2 (Beth)	73	8798
i2b2 (Partners)	97	7517
GENIA	2000	18,546

4.2 Evaluation Measures

To evaluate the performance of the system we used are three standard measures: precision, recall, and F-score. In information retrieval and pattern recognition, precision means “the ratio of the number of retrieved relevant records to the total number of relevant and irrelevant records,” or “number of true positive over number of true and false positive [98]”. And recall or sensitivity means “the ratio of the number of retrieved relevant records to the total number of relevant records,” or “number of true positive over number of true positive and number of false negative [98].” We can define precision and recall by equations (3) and (4).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{3}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{4}$$

We use two scoring schemes: strict and relaxed as they were used in SemEval [87]. The strict scoring scheme only counts perfect matches as success, a system gets zero credit if there is any extra or missing token from the correct entity to be extracted. For example, if the entity to be extracted is “congenital heart failure” and the extracted entity is “heart failure” then this is counted as failure and is given no credit. To find strict scoring, equations (3) and (4) [99] are used.

Another scoring scheme is called relaxed, which gives some credit for partially matching with the correct entity, in that precision and recall are defined as follows:

$$Precision = \frac{weighted\ True\ Positive}{True\ Positive + False\ Positive} \quad (5)$$

$$Recall = \frac{weighted\ True\ Positive}{True\ Positive + False\ Negative} \quad (6)$$

Weighted *true positive* is not simply counted as 1 (correct) and 0 (incorrect), but is assigned one of the values in Table 12. This scheme gives partial score for disjunctions [99].

Table 12 Values for weighted true positive

	Before	**Overlap**	**After**	**Before or Overlap**	**Overlap or After**	**Weight**
Before	1	0	0	0.5	0	0.33
Overlap	0	1	0	0.5	0.5	0.33
After	0	0	1	0	0.5	0.33
Before or Overlap	0.5	0.5	0	1	0.5	0.66
Overlap or After	0	0.5	0.5	0.5	1	0.66
Weight	0.33	0.33	0.33	0.67	0.67	1

Based on definitions of precision and recall F-score can be obtained by (7) as their harmonic mean:

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

4.3 Baseline for Comparison

In this section, we present results of exact matching for each dataset and for each entity class. These results have been generated by simply matching terms in the text to the terms in UMLS of the correct semantic groups and calling these terms as the extracted entities. Tables 13-15 show results of exact matching on the datasets used in our research.

Table 13 Exact Matching results on SemEval corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Disease/Disorder	0.819	0.447	0.578	0.954	0.524	0.676

Table 14 Exact Matching results on i2b2 corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Problem	0.2	0.383	0.27	0.421	0.81	0.554
Test	0.161	0.446	0.236	0.298	0.85	0.441
Treatment	0.098	0.251	0.14	0.265	0.695	0.383

Table 15 Exact Matching results on GENIA corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Protein	0.23	0.41	0.29	0.302	0.565	0.393
DNA	0.13	0.04	0.061	0.41	0.125	0.192
RNA	0.046	0.14	0.067	0.18	0.6	0.276
Cell type	0.237	0.236	0.227	0.363	0.453	0.403
Cell line	0.006	0.75	0.011	0.006	0.75	0.011

We used results in Tables 13-15 as baseline for comparing our method and Zhang's and Noemie's method. In Tables 16-18 we show results of our Unambiguous exact matching that was used to generate weakly-labeled data. Table 17 that shows unambiguous exact matching results for GENIA corpus is almost same as exact matching results listed in Table 15.

Table 16 Unambiguous Exact Matching results on SemEval corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Disease/Disorder	0.762	0.281	0.411	0.96	0.357	0.52

Table 17 Unambiguous Exact Matching results on i2b2 corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Problem	0.18	0.343	0.236	0.391	0.76	0.517
Test	0.141	0.41	0.21	0.278	0.82	0.415
Treatment	0.093	0.251	0.135	0.254	0.69	0.372

Table 18 Unambiguous Exact Matching results on GENIA corpus

Entity Class	**Strict**			**Relaxed**		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Protein	0.182	0.369	0.244	0.512	0.252	0.338
DNA	0.13	0.04	0.061	0.41	0.125	0.192
RNA	0.046	0.14	0.067	0.18	0.6	0.276
Cell type	0.237	0.236	0.227	0.363	0.453	0.403
Cell line	0.006	0.75	0.011	0.006	0.75	0.011

4.4 CRF Trained using Weakly-labeled Training Examples

In our experiment, we did train a CRF using weakly-labeled training examples to show that sequential models are not appropriate to use as machine learning tool for entity extracting in our framework of unsupervised NER using weakly-labeled data. Table 19 shows results of the CRF trained with weakly-labeled data. We did this experiment only on SemEval 2014 data set, and did not get good results as expected.

Table 19 Results of the CRF based NER system trained with weakly-labeled data

Corpus	**Strict**			**Relaxed**		
	Precision	Recall	F-score	Precision	Recall	F-score
SemEval2014	0.751	0.356	0.483	0.882	0.463	0.61

4.5 Named Entity Detection Results

The first phase of our method is entity detection. In this method, we used a classifier, trained with weakly-labeled data, for extracting named entities in different classes. As mentioned in Chapter 3, Section 3.4.4.1, there is a threshold for the classifier for extracting named entities. This threshold is different for each named entity class, and was set based on pilot studies. After some experiments and testing different thresholds for maximizing F-score, we reached the best results of named entity detection shown in Tables 20-22. Precision-recall curves of best threshold values are shown in Figures 4-8.

Table 20 Best results of Named Entity Detection on SemEval corpus

Entity Class	Strict			Relaxed		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Disease/Disorder	0.78	0.571	0.659	0.884	0.65	0.749

Table 21 Best results of Named Entity Detection on i2b2 corpus

Entity Class	Strict			Relaxed		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Problem	0.367	0.501	0.424	0.622	0.858	0.721
Test	0.251	0.321	0.281	0.456	0.585	0.512
Treatment	0.116	0.189	0.143	0.314	0.516	0.39

Table 22 Best results of Named Entity Detection on GENIA corpus

Entity Class	Strict			Relaxed		
	Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Protein	0.501	0.411	0.452	0.501	0.608	0.549
DNA	0.118	0.06	0.08	0.356	0.211	0.265
RNA	0.166	0.059	0.087	0.61	0.215	0.318
Cell type	0.362	0.223	0.276	0.405	0.621	0.49
Cell line	0.006	0.75	0.011	0.006	0.75	0.011

Other entity classes except protein did not have any improvement using our classification based named entity detector.

The best thresholds for entity classes are listed in Table 23.

Table 23 Threshold used in different corpora and entity classes

Corpus	Entity Class	Threshold Value
SemEval	Disease/Disorder	0.7
i2b2	Problem	0.6
	Test	0.6
	Treatment	0.2
GENIA	Protein	0.4
	DNA	0.2

Figures 8-12 show impact of different threshold on precision and recall on different data sets and different entity classes.

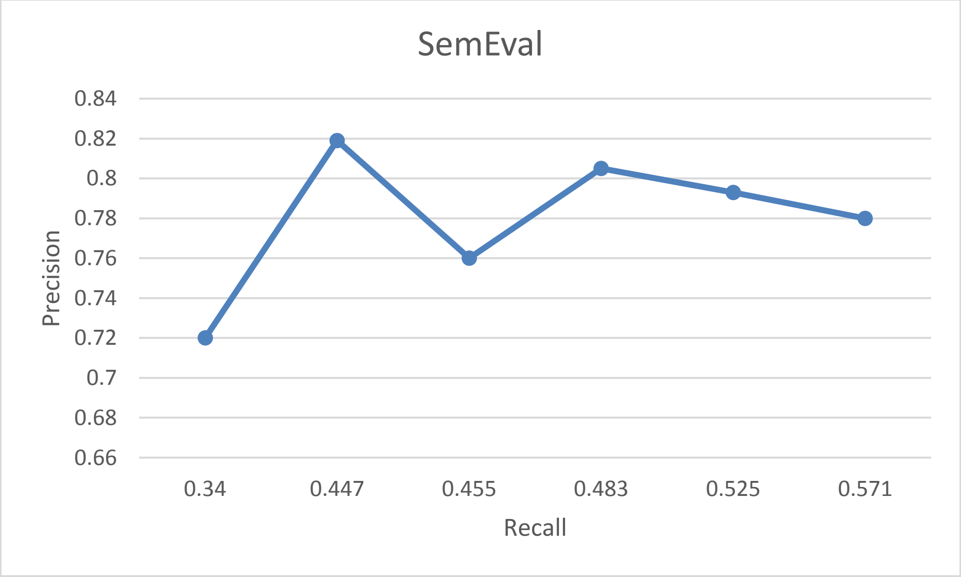


Figure 7 Precision-Recall curve on SemEval corpus

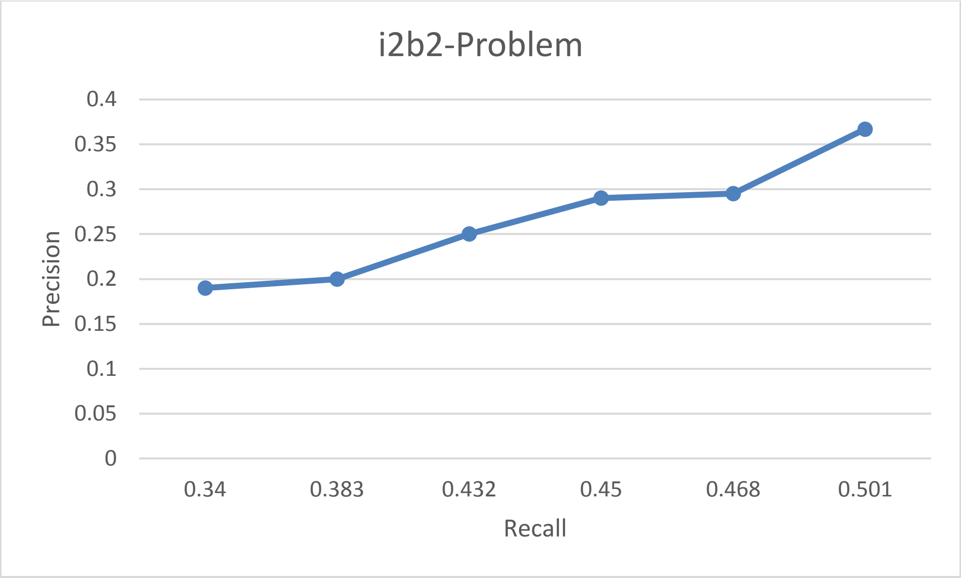


Figure 8 Precision-Recall curve on i2b2 corpus, Problem class

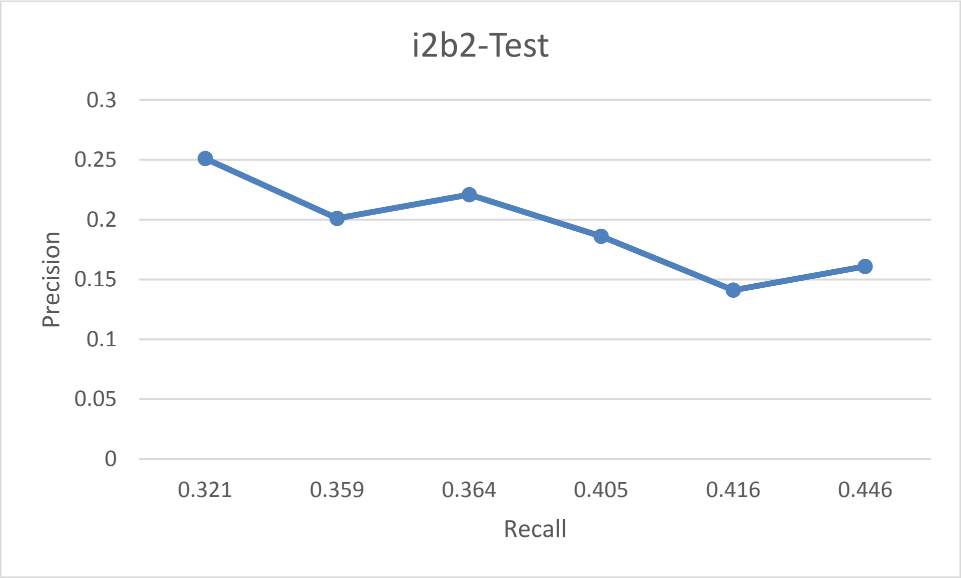


Figure 9 Precision-Recall curve on i2b2 corpus, Test class

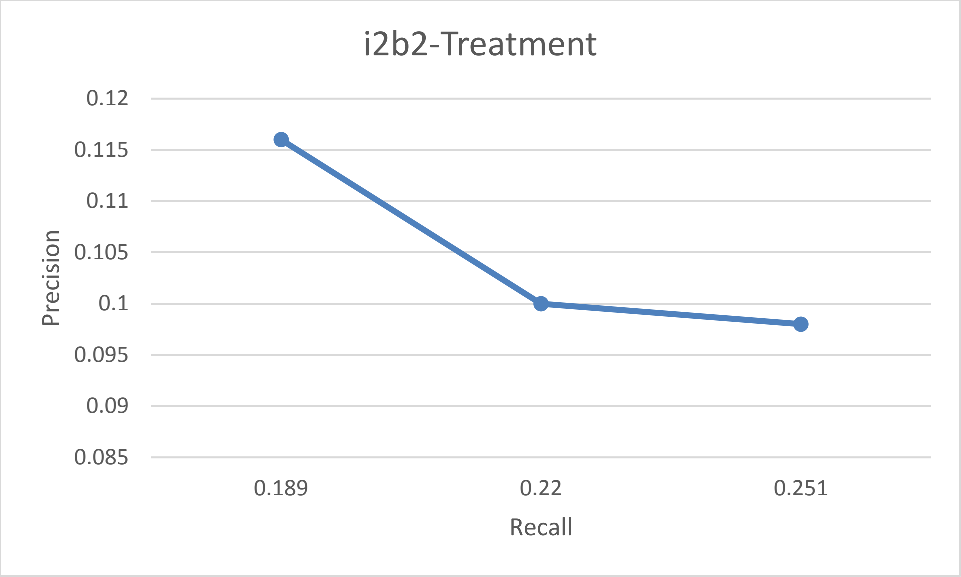


Figure 10 Precision-Recall curve on i2b2 corpus, Treatment class

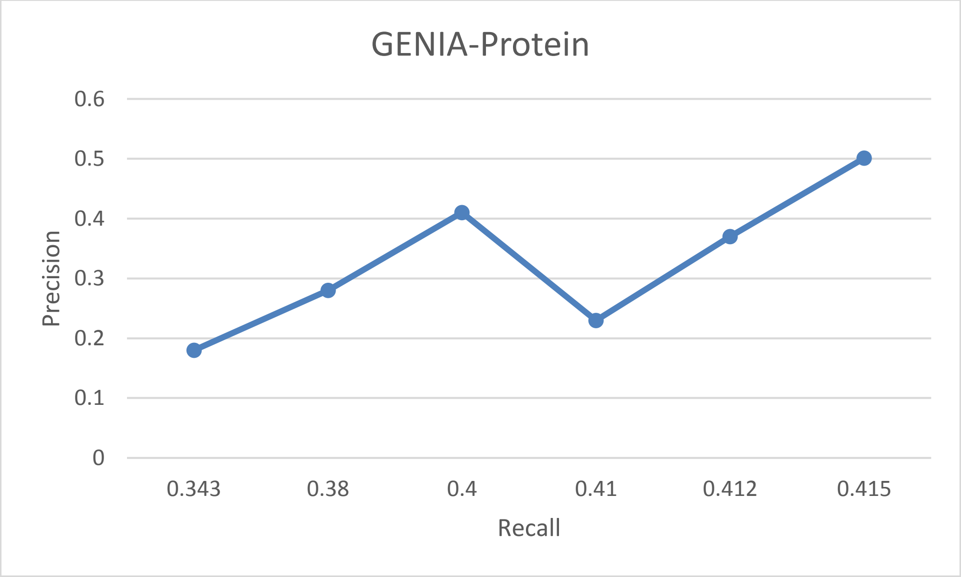


Figure 11 Precision-Recall curve on GENIA corpus, Protein class

4.6 Boundary Expansion Results

This section presents results of two methods of correcting boundaries. The first one was based on WordNet (described in Section 3.3.1) and the second one developed using a machine learning classification tool (described in Section 3.3.2). Tables 24-26 show results of improvement of our boundary expansion method on different datasets and entity classes.

Table 24 Improvement of Boundary detection methods on SemEval corpus

Entity Class	***Boundary Detection method***	**Strict**			**Relaxed**		
		Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Disease/Disorder	*No Detection*	0.78	0.571	0.659	0.884	0.65	0.749
	WordNet	0.764	0.599	0.671	0.871	0.685	0.766
	Machine learning	**0.783**	**0.622**	**0.693**	**0.881**	**0.69**	**0.773**

Table 25 Improvement of Boundary detection methods on i2b2 corpus

Entity Class	***Boundary Detection method***	**Strict**			**Relaxed**		
		Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Problem	*No Detection*	0.367	0.501	0.424	0.622	0.858	0.721
	WordNet	0.379	0.522	0.439	0.621	0.86	0.721
	Machine learning	**0.391**	**0.533**	**0.451**	**0.623**	**0.858**	**0.721**
Test	*No Detection*	0.251	0.321	0.281	0.456	0.585	0.512
	WordNet	0.265	0.329	0.293	0.456	0.586	0.512
	Machine learning	**0.284**	**0.335**	**0.307**	**0502**	**0.596**	**0.542**
Treatment	*No Detection*	0.116	0.189	0.143	0.314	0.516	0.39
	WordNet	0.116	0.189	0.143	0.314	0.516	0.39
	Machine learning	**0.129**	**0.202**	**0.156**	**0.312**	**0.519**	**0.39**

Table 26 Improvement of Boundary detection methods on GENIA corpus

Entity Class		**Strict**			**Relaxed**		
Protein	*Boundary Detection method*	*Precision*	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
	No Detection	0.501	0.411	0.452	0.501	0.608	0.549
	WordNet	0.501	0.411	0.452	0.501	0.608	0.549
	Machine learning	**0.522**	**0.43**	**0.471**	**0.522**	**0.638**	**0.574**
DNA	*No Detection*	0.118	0.06	0.08	0.356	0.211	0.265
	WordNet	0.118	0.06	0.08	0.356	0.211	0.265
	Machine learning	0.118	0.06	0.08	0.356	0.211	0.265
RNA	*No Detection*	0.166	0.059	0.087	0.61	0.215	0.318
	WordNet	0.166	0.059	0.087	0.61	0.215	0.318
	Machine learning	0.166	0.059	0.087	0.61	0.215	0.318
Cell type	*No Detection*	0.362	0.223	0.276	0.405	0.621	0.49
	WordNet	0.362	0.223	0.276	0.405	0.621	0.49
	Machine learning	0.362	0.223	0.276	0.405	0.621	0.49
Cell line	*No Detection*	0.006	0.75	0.011	0.006	0.75	0.011
	WordNet	0.006	0.75	0.011	0.006	0.75	0.011
	Machine learning	0.006	0.75	0.011	0.006	0.75	0.011

4.7 Result Comparisons

In this section, we compared results of our unsupervised method with the baseline (exact matching) and our supervised system for SemEval 2014 dataset. Furthermore, we compared our results on i2b2 and GENIA corpora with the results of the unsupervised method developed in [51].

Table 27 Performance of different methods on SemEval 2014 dataset

Entity Class	***Method***	**Strict**			**Relaxed**		
		Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Disease/Disorder	*Exact Match*	0.819	0.447	0.578	0.954	0.524	0.676
	Supervised	0.787	0.726	0.755	0.911	0.856	0.883
	Our method	**0.783**	**0.622**	**0.693**	**0.881**	**0.69**	**0.773**

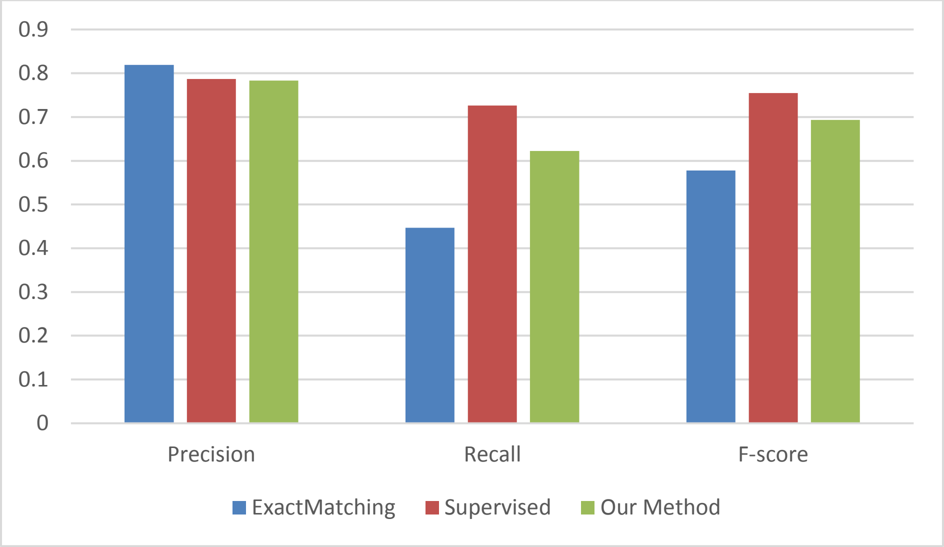


Figure 12 Comparison, SemEval 2014 corpus, Strict

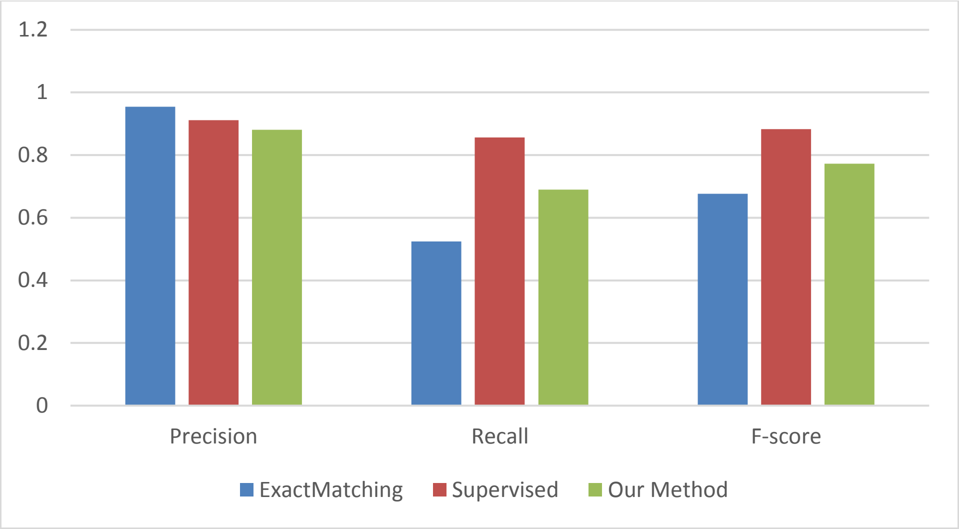


Figure 13 Comparison, SemEval 2014 corpus, Relaxed

As illustrated in Figures 13 and 14, and Table 27 our method reached high accuracy against the supervised method. The significant point is that, our method did not use any annotated text for training, and it is completely unsupervised. It is also interesting to point out that our unsupervised method, in fact, did better than many of the supervised methods that had competed in SemEval 2014 (see Chapter 2, Table 5).

For i2b2 and GENIA corpus, we also have depicted detailed comparison of different methods of NER in Tables 28 and 29.

Table 28 Comparison of different methods on i2b2 corpus

Entity Class	***Method***	**Strict**			**Relaxed**		
		Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Problem	*Exact Match*	0.2	0.383	0.27	0.421	0.81	0.554
	Zhang&Noemie's	0.267	0.317	0.291	0.492	0.715	0.583
	Our method	**0.391**	**0.533**	**0.451**	**0.623**	**0.858**	**0.721**
Test	*Exact Match*	0.161	0.446	0.236	0.298	0.85	0.441
	Zhang&Noemie's	0.369	0.221	0.277	0.546	0.526	0.536
	Our method	**0.284**	**0.335**	**0.307**	**0502**	**0.596**	**0.542**
Treatment	*Exact Match*	0.098	0.251	0.14	0.265	0.695	0.383
	Zhang&Noemie's	**0.286**	**0.159**	**0.204**	**0.454**	**0.379**	**0.413**
	Our method	0.129	0.202	0.156	0.312	0.519	0.39

Table 29 Comparison of different methods on GENIA corpus

Entity Class	***Method***	**Strict**			**Relaxed**		
		Precision	*Recall*	*F-score*	*Precision*	*Recall*	*F-score*
Protein	*Exact Match*	0.23	0.41	0.29	0.302	0.565	0.393
	Zhang&Noemie’s	0.203	0.113	0.145	0.528	0.367	0.433
	Our method	**0.522**	**0.43**	**0.471**	**0.522**	**0.638**	**0.574**
DNA	*Exact Match*	0.13	0.04	0.061	0.41	0.125	0.192
	Zhang&Noemie’s	0.056	0.091	0.069	0.3	0.532	0.384
	Our method	**0.118**	**0.06**	**0.08**	**0.356**	**0.211**	**0.265**
RNA	*Exact Match*	0.046	0.14	0.067	0.18	0.6	0.276
	Zhang&Noemie’s	**0.299**	**0.413**	**0.347**	**0.486**	**0.698**	**0.573**
	Our method	0.166	0.059	0.087	0.61	0.215	0.318
Cell type	*Exact Match*	0.237	0.236	0.227	0.363	0.453	0.403
	Zhang&Noemie’s	**0.407**	**0.367**	**0.386**	**0.504**	**0.487**	**0.495**
	Our method	0.362	0.223	0.276	0.405	0.621	0.49
Cell line	*Exact Match*	0.006	0.75	0.011	0.006	0.75	0.011
	Zhang&Noemie’s	0.05	0.118	0.071	0.128	0.33	0.185
	Our method	0.006	0.75	0.011	0.006	0.75	0.011

Figures 15-18, depicted F-score results of different entity classes in i2b2 and GENIA corpora (protein class).

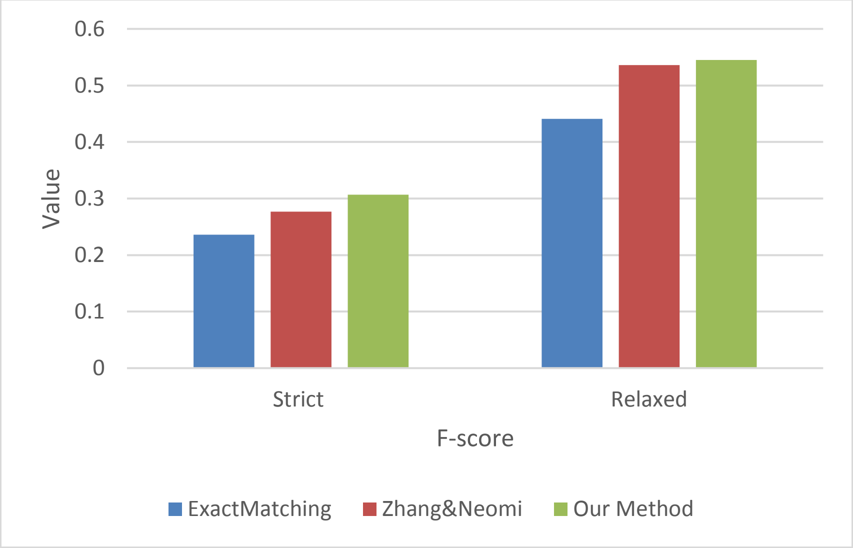


Figure 14 F-score comparison of different method on i2b2 corpus, Test class

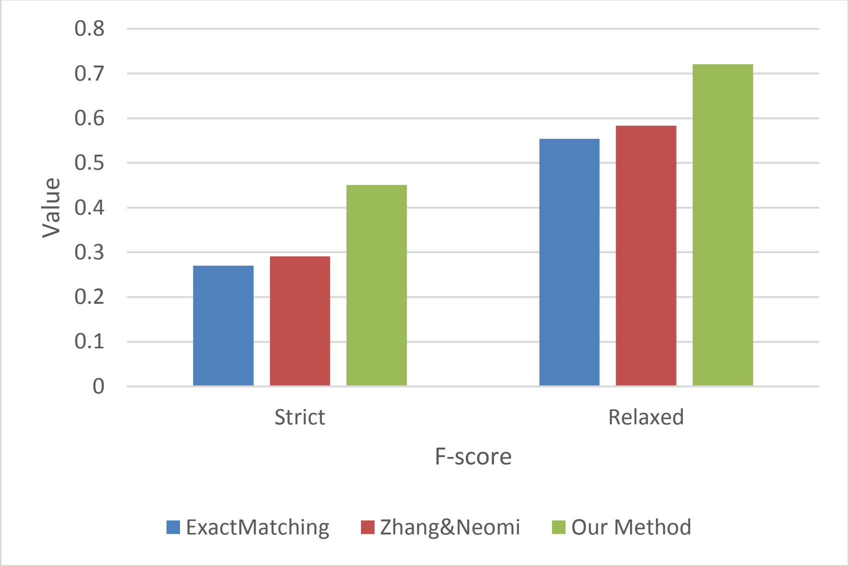


Figure 15 F-score comparison of different method on i2b2 corpus, Problem class

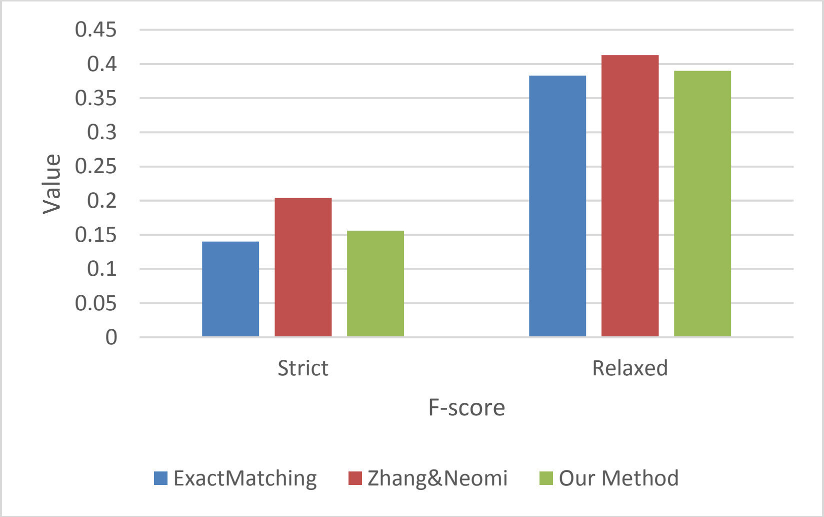


Figure 16 F-score comparison of different method on i2b2 corpus, Treatment class

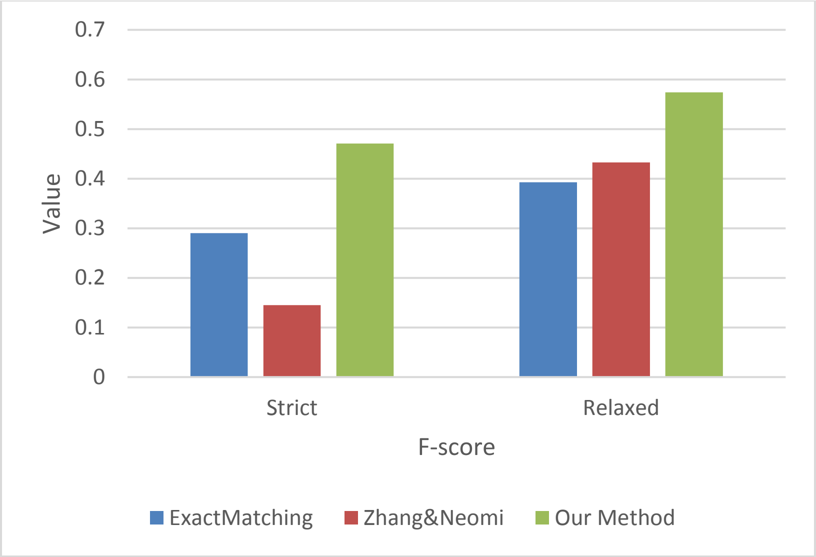


Figure 17 F-score comparison of different method on GENIA corpus, Protein class

4.8 Discussion

In this section, we have discussed how well our results are versus results generated by other methods.

In SemEval data set, we could reach a high precision while getting low recall. This means the algorithm found most of mentions correctly, thus high quality weakly-labeled data. After training the classifier, many named entities were detected. Detecting more mentions by the classifier resulted in increasing recall and decreasing precision. Finally, the most accurate results were acquired when a balance established between precision and recall. Although, we could not reach the supervised system accuracy, but results show the system is reliable and robust without using any manually annotated training data. For detecting names of diseases/disorders in SemEval 2014 corpus, we used 12 semantic types included in “Disorder” semantic group in UMLS. This semantic group has 398,725 terms, which gives us a rich resource named entities. More details on named entity classes are presented in Table 30.

Furthermore, results on i2b2 data set, problem and test entity classes, show our system has performed more accurate than the unsupervised NER developed by Zhang and Noemie in [51]. Moreover, our developed biomedical NER could reach an F-score at least three times better than the results presented in [51] by Zhang in Protein entity class on GENIA corpus. The main reason is that we used a machine learning tool for detecting mentions trained with high quality data, generated by unambiguous exact matching. Unambiguous exact matching algorithm considers those mentions with unique semantic types, thus creating very accurate positive and negative examples. Training the classifier with highly precise data causes detecting desired named entities well.

Another reason for reaching better performance of our method versus Zhang's and Noemie's method is that each named entity class has a specific list of named entities, richer the named entity resources, better performance of the NER. We can see that in problem, test, and protein entity classes. In Table 24 details of named entities from UMLS used in our approach are presented.

In other named entity class, DNA, we reached higher accuracy but not too much, and in other classes RNA, Cell type, and Cell line Zhang's and Noemie's method generated more accurate results. The main reason of not reaching high accuracy in those classes is we lack specific lists of named entities. Zhang and Noemie used some UMLS concepts, listed in Table 24, instead of using a list of terms in semantic type or group. These concepts were used for detecting named entities in DNA, RNA, Cell type, and Cell line classes. On the other hand, our developed method needed a specific list of named entities to generate weakly-labeled data. Therefore, our system has better performance on named entity classes that have particular list of names, and less improvement and/or low accuracy on those that there is not any special catalog of names. This also affects the second classifier for boundary expansion. This classifier uses terms in UMLS for training, therefore more terms, more accurate boundary detection. As seen in tables 18-20, the boundary detection method could improve the accuracy in classes with specific list of names entities, problem, test, treatment, and protein classes, but in other named entity classes it did not have any impact for improving accuracy.

Table 30 List of named entities and semantic types used in each class

Data set	Entity class	Domain used in Zhang and Noemie's method	Domain used in our method	Number of terms	
				Zhang' method	Our method
SemEval	Disease/Disorder	N/A	Disorders (semantic group)	N/A	860k
i2b2	Problem	Disorders (semantic group)	Disorders (semantic group)	398,725	860k
	Test	Laboratory Procedure + Laboratory or Test Result + Diagnostic Procedure semantic types	Laboratory Procedure + Laboratory or Test Result + Diagnostic Procedure semantic types	66,015	210k
	Treatment	Therapeutic or Preventive Procedure + Clinical Drug semantic types	Therapeutic or Preventive Procedure and Clinical Drug semantic groups	153,084	1.3m
GENIA	Protein	Amino Acid, Peptide, or Protein semantic type	Amino Acid, Peptide, or Protein semantic type	35,351	390k
	DNA	C0012854 (DNA, Desoxyribonucleic acid)	Biologically Active Substance semantic type	45,671	296k
	RNA	C0035668 (ALT, Ribonucleic acid)	Nucleic Acid, Nucleoside, or Nucleotide	1,029	449k

			semantic type		
	Cell type	C0007600 (Cultured Cell Line)	Cell semantic type	423	29k
	Cell line	C0449475 (Cell type)	Intellectual Product semantic type	264,729	52k

The only class that our system could not reach better accuracy than Zhang's and Noemie's system is treatment class, even though there is a particular list of named entities. We expect that the reason is because of lack of standard extra clinical notes for training the classifier. For class disease/disorder in SemEval or problem in i2b2 data sets, we used extra 5000 of clinical notes to train the classifier. But in classes treatment and test in i2b2 data set and protein, DNA, RNA, cell type, and cell line in GENIA corpus, we only used training sets provided in the databases.

5 Future Work

In future, there is possibility to implement is a semi-supervised system using manually annotated training data and generated data by our method to beat supervised system that only use annotated training data. In a semi-supervised system, named entity detection classifier outputs and a set of manually annotated training data can be used to train a semi-supervised system. This kind of system may beat a supervised system that is trained with only manually annotated text, because the data generated by the unsupervised part may have information missing from the manually annotated data. Therefore, developing such a system will be an important step for developing a highly accurate named entity recognition system.

The second aim in future might be using the unsupervised biomedical NER for other languages such as French and Italian. There are good resources of manually annotated text in different domains in English, but other languages usually do not have such resources. One of the main problem with supervised systems is that they cannot be extended to other languages and furthermore other named entity classes. Thus, unsupervised methods are appropriate solution to extract named entities from free text in other languages, and also in different domains. For instance, France is a leading country in embedding natural language processing techniques in healthcare, but they do not have rich resources of annotated text to develop supervised systems. Furthermore, I would like to develop an unsupervised system for detecting named entities in Farsi, language spoken in Iran. This kind of system might be challenging because of difference of alphabet, direction of writing which is right to left, lack of basic tools NLP such as POS tagger, Stemmer, Tokenizer, etc..

Fortunately using information extraction techniques is rapidly growing across the world and most of countries need such systems not only to use in their healthcare systems, but also in other domains such as Economy, Agriculture, Art, History, and other sciences.

6 Conclusion

Starting point of this research was to achieve the aim of reducing manually annotated text for task of named entity recognition in biological and medical domains. In our project, we presented a novel method for extracting named entities from biomedical text without the aid of annotated data sets. In this research, we showed that it is possible to reach highly accurate results without using any kind of manually annotated text. This success was because of using machine learning trained with high quality weakly-labeled data. Our developed system had more accurate results than the other unsupervised named entity recognition system.

Another aspect that we addressed in our research is that the developed method did perform well on data from different domains. Results on different data sets and different named entity classes showed that our system is reliable and robust on other datasets and entity classes if provided with a list of entity names with their semantic types and a large unannotated corpus. Our developed NER system, as shown in results, is robust and reliable given a list of entity names with their semantic types and a large unannotated corpus. Thus, it can be used to detect named entities in other classes and furthermore can be applied to other types of text in English and other languages.

Finally, to get access to the tool that we developed in this research, please go to the following link:

- https://sites.google.com/view/oghiasvand/

If the link did not work, please contact me at o.ghiasvand@gmail.com to get the tool.

7 References

[1] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3-26.

[2] Tanabe, L., Xie, N., Thom, L. H., Matten, W., & Wilbur, W. J. (2005). GENETAG: A tagged corpus for gene/protein named entity recognition. BMC bioinformatics, 6(Suppl 1), S3.

[3] Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of HLT-NAACL, pp. 142-147. Association of Computational Linguistics.

[4] Pradhan S., Elhadad N., Chapman W., Manandhar S., & Savova G. (2014). SemEval-2014 task 7: Analysis of clinical text. In Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014), pp. 54-62.

[5] Karamanis, N., Lewin, I., Seal, R. L., Drysdale, R. A., & Briscoe, E. J. (2007). Integrating natural language processing with FlyBase curation. In Pacific symposium on Biocomputing, Vol. 12, pp. 245-256.

[6] Hoffmann, R., & Valencia, A. (2005). Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics, 21(suppl 2), 252-258.

[7] Zelenko, D., Aone, C., & Richardella, A. (2003). Kernel methods for relation extraction. The Journal of Machine Learning Research, 3, 1083-1106.

[8] Mallory, E. K., Zhang, C., Ré, C., & Altman, R. B. (2015). Large-scale extraction of gene interactions from full-text literature using DeepDive. Bioinformatics, btv476.

[9] Cardie, C. (1997). Empirical methods in information extraction. AI magazine, 18(4), 65.

[10] Cowie, J., & Lehnert, W. (1996). Information extraction. Communications of the ACM, 39(1), 80-91.

[11] Ramani, A. K., Bunescu, R. C., Mooney, R. J., & Marcotte, E. M. (2005). Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. Genome biology, 6(5), R40.

[12] "Mining the Biomedical Literature","publisher":"The MIT Press","publisher place":"Cambridge,Mass","number-of-pages":"150","edition":"1st edition","source":"Amazon","event-place":"Cambridge, Mass","ISBN":"978-0-262-01769-5".

[13] Fukuda, K. I., Tsunoda, T., Tamura, A., & Takagi, T. (1998). Toward information extraction: Identifying protein names from biological papers. In Pac Symp Biocomput, 707(18), pp. 707-718.

[14] 1Muslea, I. (1999). Extraction patterns for information extraction tasks: A survey. In The AAAI-99 Workshop on Machine Learning for Information Extraction (Vol. 2, No. 2).

[15] Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, Wong YW (2005). Comparative experiments on learning information extractors for proteins and their interactions. Artificial intelligence in medicine, 33(2), 139-155.

[16] Peng, F., & McCallum, A. (2006). Information extraction from research papers using conditional random fields. Information processing & management, 42(4), 963-979.

[17] Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In Proc. of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, pp. 104-107.

[18] UMLS Reference Manual, US National Library of Medicine, Bethesda, USA, 2009.

[19] https://en.wikipedia.org/wiki/Text_annotation

[20] Wolfe, Joanna; Christine M. Neuwirth (2001). "From the Margins to the Center: The Future of Annotation". Journal of Business and Technical Communication 15 (33): 333–370.

[21] "Mining the Biomedical Literature","publisher":"The MIT Press","publisher place":"Cambridge,Mass","number-of-pages":"150","edition":"1st edition" ,"source": "Amazon", "event-place": "Cambridge, Mass","ISBN":"978-0-262-01769-5".

[22] https://www.nlm.nih.gov/research/umls/

[23] Charles Sutton, Andrew McCallum, An Introduction to Conditional Random Fields, Foundations and Trends in sample, ol. 4, No. 4 pages 267–373, 2011.

[24] Fosler Lussier, Eric, Markov Models and Hidden Markov Models, International computer science institute, California, USA, December 1998.

[25] N Cristianini, J Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, US, 2000.

[26] Friedman C, Alderson P, Austin J, Cimino J, Johnson S. A general naturallanguage text processor for clinical radiology. J Am Med Inform Assoc1994;1(2):161–74.

[27] Tanabe L, Wilbur W. Tagging gene and protein names in biomedical text. Bioinformatics 2002;18(8):1124–32.

[28] Fukuda K, Tsunoda T, Tamura A, Takagi T, et al. Toward information extraction: identifying protein names from biological papers. In: Pac symp biocomput, vol. 707; 1998. p. 707–18.

[29] Proux D, Rechenmann F, Julliard L, Pillet V, Jacq B, et al. Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. Genome informatics series 1998:72–80.

[30] Rindflesch T, Tanabe L, Weinstein J, Hunter L. Edgar: extraction of drugs, genes and relations from the biomedical literature. In: Pacific symposium on biocomputing. NIH public access; 2000. p. 517.

[31] Ratinov L, Roth D. Design challenges and misconceptions in named entity recognition. In: Proceedings of the thirteenth conference on computational natural language learning. Association for Computational Linguistics; 2009. p. 147–55.

[32] Friedman C, Kra P, Yu H, Krauthammer M, Rzhetsky A. Genies: a natural language processing system for the extraction of molecular pathways from journal articles. Bioinformatics 2001;17(Suppl. 1):S74–82.

[33] Aronson A. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In: Proceedings of the AMIA symposium. American Medical Informatics Association; 2001. p. 17.

[34] Bodenreider O. The unified medical language system (umls): integrating biomedical terminology. Nucleic Acids Res 2004;32(Suppl. 1):D267–70.

[35] Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, et al. Gene ontology: tool for the unification of biology. Nat Genet 2000;25(1):25.

[36] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J. Am Med Inform Assoc 2010;17(5):507–13.

[37] Jayan J, R Rajeev, Sherly E, A Hybrid Statistical Approach for Named Entity Recognition for Malayalam Language, International Joint Conference on Natural Language Processing, pages 58–63, Japan, October 2013.

[38] Kim J, Ohta T, Tateisi Y, Tsujii J. Genia corpusa semantically annotated corpus for bio-textmining. Bioinformatics 2003;19(Suppl. 1):i180–2.

[39] GuoDong Z, Jian S. Exploring deep knowledge resources in biomedical name recognition. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics; 2004. p. 96–9.

[40] Kazama J, Makino T, Ohta Y, Tsujii J. Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the ACL-02 workshop on natural language processing in the biomedical domain-volume 3. Association for Computational Linguistics; 2002. p. 1–8.

[41] Mitsumori T, Fation S, Murata M, Doi K, Doi H. Gene/protein name recognition based on support vector machine using dictionary as features. BMC Bioinformatics 2005.

[42] Wang Y, Patrick J. Cascading classifiers for named entity recognition in clinical notes. In: Proceedings of the workshop on biomedical information extraction. Association for Computational Linguistics; 2009. p. 42–9.

[43] Settles B. Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics; 2004. p. 104–7.

[44] McDonald R, Pereira F. Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics 2005;6(Suppl. 1):S6.

[45] Kim J, Ohta T, Tsuruoka Y, Tateisi Y, Collier N. introduction to the bio-entity recognition task at jnlpba. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications; 2004. p. 70–5.

[46] Omid Ghiasvand, Rohit J. Kate. 2014. UWM: Disorder mention extraction from clinical text using crfs and normalization using learned edit distance patterns. In Proceedings of the International Workshop on Semantic Evaluations, Dublin, Ireland, August 2014.

[47] Omid Ghiasvand, Disease Name Extraction from Clinical Text Using Conditional Random Fields, Master's Thesis, University of Wisconsin-Milwaukee, May 2014.

[48] Omid Ghiasvand, Rohit Kate, UWM: A Simple Baseline Method for Identifying Attributes of Disease and Disorder Mentions in Clinical Text, SemEval 2015, pages 385-388, Texas, USA, August 2015.

[49] Yeh A, Morgan A, Colosimo M, Hirschman L. Biocreative task 1a: gene mention finding evaluation. BMC Bioinformatics 2005;6(Suppl. 1):S2.

[50] Hirschman L, Yeh A, Blaschke C, Valencia A. Overview of biocreative: critical assessment of information extraction for biology. BMC bioinformatics 2005.

[51] Shaodian Zhang and Noemi Elhadad, Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. Journal of Biomedical Informatics, Volume 46, Issue 6, December 2013, Pages 1088-1098.

[52] Solt I, Gerner M, Thomas P, Nenadic G, Bergman CM, Leser U, et al. Gene mention normalization in full texts using gnat and linnaeus. In: Proceedings of the BioCreative III workshop (Bethesda, USA); 2010. p. 134–9.

[53] Kim J-D, Ohta T, Pyysalo S, Kano Y, Tsujii J. Overview of bionlp'09 shared task on event extraction. In: Proceedings of the workshop on current trends in biomedical natural language processing: shared task. Association for Computational Linguistics; 2009. p. 1–9.

[54] Uzuner Ö, South B, Shen S, DuVall S. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc 2011;18(5):552–6.

[55] Meystre S, Haug P. Comparing natural language processing tools to extract medical problems from narrative text. In: AMIA annual symposium proceedings, vol. 2005. American Medical Informatics Association; 2005. p. 525.

[56] Abacha A, Zweigenbaum P. Medical entity recognition: a comparison of semantic and statistical methods. In: Proceedings of BioNLP 2011 workshop. Association for Computational Linguistics; 2011. p. 56–64.

[57] Abacha A, Zweigenbaum P. Automatic extraction of semantic relations between medical entities: a rule based approach. J Biomed Semant 2011;2(Suppl. 5):S4.

[58] Patrick P, Wang Y, Budd P. Automatic mapping clinical notes to medical terminologies. In: Australasian language technology workshop; 2006. p. 75.

[59] Wang Y. Annotating and recognising named entities in clinical notes. In: Proceedings of the ACL-IJCNLP 2009 student research workshop. Association for Computational Linguistics; 2009. p. 18–26.

[60] Wang Y, Patrick J. Cascading classifiers for named entity recognition in clinical notes. In: Proceedings of the workshop on biomedical information extraction. Association for Computational Linguistics; 2009. p. 42–9.

[61] Kang N, Afzal Z, Singh B, Van Mulligen EM, Kors JA. Using an ensemble system to improve concept extraction from clinical records. J Biomed Inform 2012;45(3):423–8.

[62] De Bruijn B, Cherry C, Kiritchenko S, Martin J, Zhu X. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. J Am Med Inform Assoc 2011;18(5):557–62.

[63] Tang B, Cao H, Wu Y, Jiang M, Xu H. Clinical entity recognition using structural support vector machines with rich features. In: Proceedings of the ACM sixth international workshop on data and text mining in biomedical informatics. ACM; 2012. p. 13–20.

[64] S. V. Ramanan and P. Senthil Nathan. 2014. RelAgent: Entity detection and normalization for diseases in clinical records: a linguistically driven approach. In Proceedings of the International Workshop on Semantic Evaluations, Dublin, Ireland, August 2014

[65] Sergio Matos, Tiago Nunes, and Jose Luıs Oliveira. 2014. BioinformaticsUA: Concept recognition in clinical narratives using a modular and highly ef- ficient text processing framework. In Proceedings of the International Workshop on Semantic Evaluations, Dublin, Ireland, August 2014.

[66] Suwisa Kaewphan, Kai Hakaka1, and Filip Ginter. 2014. UTU: Disease mention recognition and normalization with crfs and vector space representations. In Proceedings of the International Workshop on Semantic Evaluations, Dublin, Ireland, August 2014.

[67] Yaoyun Zhang, Jingqi Wang, Buzhou Tang, Yonghui Wu, Min Jiang, Yukun Chen, and Hua Xu. 2014. UTH CCB: A report for SemEval 2014 task 7 analysis of clinical text. In Proceedings of the International Workshop on Semantic Evaluations, Dublin, Ireland, August 2014.

[68] Rau L. Extracting company names from text. In: Proceedings of the seventh IEEE conference on artificial intelligence applications, vol. 1. IEEE; 1991. p. 29–32.

[69] Coates-Stephens S. The analysis and acquisition of proper names for the understanding of free text. Comput Humanit 1992;26(5):441–56.

[70] Fellbaum C. Wordnet. Theory and applications of ontology: computer applications 2010:231–43.

[71] Alfonseca E, Manandhar S. An unsupervised method for general named entity recognition and automated concept discovery. In: Proceedings of the 1st international conference on general WordNet, Mysore, India; 2002. p. 34–43.

[72] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the www. In In Proceedings of the Ontology Learning Workshop, ECAI, Berlin, Germany, 2000.

[73] Nadeau D, Turney P, Matwin S. Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. Advances in artificial intelligence 2006:266–77.

[74] Sekine S, Nobata C. Definition, dictionaries and tagger for extended named entity hierarchy. In: Proceedings of the language resources and evaluation conference (LREC); 2004. p. 1977–80.

[75] Shinyama Y, Sekine S. Named entity discovery using comparable news articles. In: Proceedings of the 20th international conference on computational linguistics. Association for Computational Linguistics; 2004. p. 848.

[76] Powers, David M W. Applications and explanations of Zipf's law. Association for Computational Linguistics: 151–160, 1998.

[77] Collins M, Singer Y. Unsupervised models for named entity classification. In: Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora; 1999. p. 100–10.

[78] Ming Xue; Changjun Zhu (25 April 2009). A Study and Application on Machine Learning of Artificial Intellligence. Artificial Intelligence, 2009. JCAI '09. International Joint Conference on. p. 272-274. doi:10.1109/JCAI.2009.55

[79] Riloff E, Jones R, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the national conference on artificial intelligence. John Wiley & Sons Ltd.; 1999. p. 474–9.

[80] Cucchiarelli A, Velardi P. Unsupervised named entity recognition using syntactic and semantic contextual evidence. Comput Linguist 2001;27(1):123–31.

[81] Etzioni O, Cafarella M, Downey D, Popescu A, Shaked T, Soderland S, et al. Unsupervised named-entity extraction from the web: an experimental study. Artif Intell 2005;165(1):91–134.

[82] Elsner M, Charniak E, Johnson M. Structured generative models for unsupervised named-entity clustering. In: Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the Association for Computational Linguistics. Association for Computational Linguistics; 2009. p. 164–72.

[83] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd annual meeting on association for computational linguistics. Association for Computational Linguistics; 1995. p. 189–96.

[84] Jurafsky D, Martin J, Kehler A. Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition, vol. 2. MIT Press; 2002.

[85] Noémie Elhadad, Shaodian Zhang, Patricia Driscoll, and Samuel Brody, Characterizing the Sublanguage of Online Breast Cancer Forums for Medications, Symptoms, and Emotions, AMIA Annu Symp Proc. 2014; 2014: 516–525, 2014.

[86] Sun Kim, Zhiyong Lu, W John Wilbur, identifying named entities from PubMed® for enriching semantic categories, BioMedCentral, DOI 10.1186/s12859-015-0487-2, 2015.

[87] Sameer Pradhan, Noemie Elhadad, B South, David, Martinez, Lee Christensen, Amy Vogel, Hanna, Suominen, W Chapman, and Guergana Savova. 2013. Task 1: ShARe/CLEF eHealth Evaluation, Lab 2013. Online Working Notes of CLEF, CLEF, 230.

[88] Charles Sutton, Andrew McCallum, An Introduction to Conditional Random Fields, Foundations and Trends in sample, ol. 4, No. 4 pages 267–373, 2011.

[89] Asif Ekbal, Sivaji Bandyopadhyay, A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi, Linguistic Issues in Language Technology, Volume 2, Issue 1, 2009.

[90] Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[91] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

[92] Rokach, Lior; Maimon, O. (2008). Data mining with decision trees: theory and applications. World Scientific Pub Co Inc. ISBN 978-9812771711.

[93] Ho, Tin Kam (1998). "The Random Subspace Method for Constructing Decision Forests" . IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi: 10.1109/34.709601.

[94] https://en.wikipedia.org/wiki/Random_subspace_method

[95] https://wordnet.princeton.edu/wordnet/

[96] https://www.elastic.co/products/elasticsearch

[97] "DB-Engines Ranking - popularity ranking of search engines". db-engines.com. Retrieved 10 January 2016.

[98] Jesse Davis, Mark Goadrich, The Relationship between Precision-Recall and ROC-Curves, Proceeding of 23rd international conference, on machine learning, Pittsburgh, PA 2006.

[99] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple,Graham Katz, James Pustejovsky, SemEval-2007 Task 15: TempEval Temporal Relation Identification, Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007) , pages 75–80, Prague, June 2007.

Curriculum Vitae

Omid Ghiasvand

Education

August 2017	**PhD in Biomedical and Health Informatics** College of Engineering and Applied Science, University of Wisconsin-Milwaukee, Milwaukee, United States of America **Dissertation: Unsupervised Biomedical Named Entity Recognition**
May 2014	**M.Sc. in Bioengineering-Medical Informatics** College of Engineering and Applied Science, University of Wisconsin-Milwaukee, Milwaukee, United States of America **Thesis: Disease Name Extraction Using Conditional Random Fields**
September 2010	**M.Sc. in Computer Engineering-Artificial Intelligence** Department of Computer Science, Science and Research Branch, Islamic Azad University, Tehran, Iran **Thesis: Time Series Prediction-Wind Speed Forecast at Manjil Wind Power Plant**
September 2007	**B.Sc. in Computer Engineering** Department of Computer Science, Islamic Azad University, Hamedan Branch, Hamedan, Iran, and Shamsipour Technical College, Tehran, Iran **Project: Robotic Object Tracker**

Positions

- **Medical College of Wisconsin**
  - ✓ **Research Associate II, since May 2015**
- **University of Wisconsin-Milwaukee**
  - ✓ **Research Assistant Fall 2011, Spring and Summer 2012**
  - ✓ **Graduate Teaching Assistant, Fall 2012- Spring 2015**

Research Interests

- **Specialties**
  - ✓ **Natural Language Processing**
  - ✓ **Text Mining**
  - ✓ **Machine Learning**
- **Data Analysis**
  - ✓ **Data Mining.**
  - ✓ **Statistical and Mathematical Modeling.**

Programming Languages

- Java (Expert)
- C/C++ (Expert)
- Matlab (Expert)
- Visual C++.Net (Familiar)
- PHP (Familiar)

Technical Skills

- Big Data Mining
  - Hadoop
    - HBase
    - Map/Reduce
  - Solr/Elastic Search Indexing

Publications

1. **Omid Ghiasvand**, Mary Shimoyama, “**Introducing a Text Annotation Tool (OntoMate), Assisting Curation at Rat Genome Database”**, Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 465-465, Seattle, USA, 2016.
2. Mary Shimoyama, Stanley JF Laulederkind, Jeff De Pons, Rajni Nigam, Jennifer R Smith, Marek Tutaj, Victoria Petri, G Thomas Hayman, Shur-Jen Wang, **Omid Ghiasvand**, Jyothi Thota, Melinda R Dwinell, **“Exploring human disease using the Rat Genome Database”**, Disease Models & Mechanisms 2016 9: 1089-1095; doi: 10.1242/dmm.026021, October 2016.
3. **Omid Ghiasvand**, Rohit J. Kate, **“A Simple Baseline Method for Identifying Attributes of Disease and Disorder Mentions in Clinical Text”** In proceeding of the International Workshop on Semantic Evalualtions, Denver, USA, June 2015.
4. **Omid Ghiasvand**, Rohit J. Kate, “**Disorder Mention Extraction from Clinical Text Using CRFs and Normalization Using Learned Edit Distance Patterns,”** In proceeding of the International Workshop on Semantic Evalualtions, Dublin, Ireland, August 2014.
5. Atish Sinha, Hemant Jain, **Omid Ghiasvand**, Carmelo Gaudioso, “**A Text Mining-Based System for Clustering Clinical Trials and Matching Patients,”** International Symposium of Information Systems, Goa, India, January 2013.
6. Amir Ghiasvand, **Omid Ghiasvand**, Zahra Moslehi, **" A Fuzzy Cross-layer based method to Improve the Quality of Service in Uplink Multimedia Transmissions in WiMAX",** 1st International Conference on Computing and Information Technology, Madina, KSA, March 2012.
7. Omid Ghiasvand**, Amir Ghiasvand,** **"**Wind Speed Short Term Forecasting by Neuro Fuzzy Modeling with Aid of Mutual Information at Manjil Wind Power Plant**," International eConference on Computer and Knowledge Engineering (IEEE IECCKE 2011), Mashahad, Iran, September 2011.**

8. **Amir Ghiasvand**, Omid Ghiasvand, "A Fuzzy Cross-layer based method to Improve the Quality of Service in Downlink Multimedia Transmissions in WiMAX," **15th Iranian International Conference of Computer (CSICC2010), Tehran, Iran, February 2010, (in Farsi)**
9. Omid Ghiasvand, **Amir Ghiasvand, Mohammad Shiri**, "Intelligent Multi Agent Soccer Coaching Adviser System**," 12th Iranian Student Conference on Electronic Engineering (ISCEE2009), Tabriz, Iran, July 2009, (in Farsi).**
10. Omid Ghiasvand**, Caro Lucas, Ahmad Kalhor, Mohammad_Reza Feizi_Derakhshi**، "Using fuzzy neural network to predict arrival time of interplanetary shocks**", Forecasting of the Radiation and Geomagnetic Storms by Networks of Particle Detectors (FORGES2008), Yerevan, Armenia, September 29-October 3, 2008.**
11. **Omid Ghiasvand**, Rohit J Kate, **Disease Name Extraction from Clinical Text Using Conditional Random Fields**, Biomedical and Health Informatics Research Institute, Informatics Day, Milwaukee, USA, May 2014.
12. Emmanuel Asante-Asamani, **Omid Ghiasvand**, Matthew Crawford, Michyio Yamada, Jugal Ghorai, Peter J. Tonellato, **"An Optimal Approach to Identifying Causal Relationships in Complex Data Sets Using Bayesian Networks,"** College of Health Sciences, Proceeding of 2012 Spring Research Symposium, Milwaukee, USA, May 2012.

Projects

1. Wind speed forecast at Manjil Wind Power Plant**, Gilan, Iran, September 2009-September 2010 (Publications 5).**
2. Breast Cancer risk factor prediction**, Milwaukee, USA, September 2011- August 2012 (Publication 10).**
3. Clinical Trials and Patients Matching**, Milwaukee, USA, June 2012-December 2013 (Publication 3).**
4. Biomedical Named Entity Recognition**, Milwaukee, USA, since January 2014 (Publications 1, 2 and 9).**
5. OntoMate, Automatic Annotating of PubMed Articles, **Rat Genome Database, Milwaukee, USA, since May 2015.**
6. Pathway Portal, **Rat Genome Database, Milwaukee, USA since October 2016.**

Honors and Awards

- CTSA Fellowship, Medical college of Wisconsin, September 2014.
- College of Engineering and Applied Science Dean's Scholarship, University of Wisconsin-Milwaukee, January 2014.
- Rand 2nd in Task 7B and 3rd in Task 7A in SemEval 2014 NLP challenge among 20 teams.
- Timothy B Patrick's Award, University of Wisconsin-Milwaukee, May 2014.
- Biomedical and Healthcare Informatics Research Institute (BHIRI) Graduate Student Research Award, May 2013.
- Chancellor's Award, University of Wisconsin-Milwaukee, January and September 2011.
- Member of the National Society of Collegiate Scholars, since 2012.
- Member of American Medical Informatics Association (AMIA), since 2012.

Teaching Experiences

- University of Wisconsin-Milwaukee

- ✓ **Introduction to Programming with JAVA, since Fall 2013.**
- ✓ **Computer Organization and Assembly Language Programming, Fall 2012 and Spring 2013.**

- Islamic Azad University
  - ✓ **Data Structures, Fall 2007 to Spring 2011.**
  - ✓ **Advanced Programming (C/C++), Fall 2007 to Spring 2011.**
  - ✓ **Object Oriented Programming Fall 2010, Spring 2011.**

Services

- Reviewer
  - ✓ **The 2010 International Conference on Informatics, Cybernetics, and Computer Applications (ICICCA 2010), Bangalore, India, July 2010.**
  - ✓ **First International Conference on Integrated Intelligent Computing (ICIIC 2010), IEEE, Bangalore, India, August 2010.**
  - ✓ **International eConference on Computer and Knowledge Engineering (IEEE IECCKE 2011), Mashahad, Iran, since September 2011.**
  - ✓ **International Workshop on Semantic Evaluations, Denver, USA, February 2015.**

.