Universiteit Leiden
Opleiding Informatica

Medical Entity Recognition

from Patient Forum Data

Name: Yuan Yiyu

Date: August 10, 2017

1st supervisor: Suzan Verberne
2nd supervisor: Wessel Kraaij

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

With the development of medical information technology, a large number of electronic health records are available. This data is not only used for clinical decision support systems, but also for medical studies. Medical entity recognition (MER) techniques have been developed to extract medical instances from these data. Nowadays, the techniques are not only used for extracting information from medical and clinical reports, but also from medical social media. However, there is a limited process regarding to MER from user narratives. Unlike structured medical records and reports, text from online forums increases more challenges due to the informal non-standard use of language. To address the issues, we aim to propose an approach for medical entity recognition from patient forum data in this paper. In our approach, we firstly utilize Unified Medical Language System (UMLS) to realize unsupervised UMLS database lookup. Beside unigram UMLS lookup, we also perform n-gram noun phrase chunking to detect multi-word medical terms. To further improve the performance, we combine the UMLS lookup results with word embedding clustering generated from the Word2vec tool and k-means clustering as input features for a supervised conditional random fields (CRF) model. Both unsupervised and supervised methods have achieved competitive results, with f-measures of 82.79% and 91.61% respectively. Our results prove that part-of-speech tags and n-gram noun phrase chunking can make great improvements over the unigram UMLS lookup baseline, and the supervised CRF method with right combination of features is necessary to further achieve a better performance in comparison with the unsupervised UMLS lookup method.

Acknowledgements

First of all, I would like to thank my supervisors, Suzan Verberne and Wessel Kraaij. Wessel offered me the opportunity to write this thesis in the first place, and gave me advice for selecting topics. Suzan gave me support and encouragement throughout the entire thesis working process. She was patient and kind for instructing and explaining related knowledge of the thesis topic. She also gave me chances and suggestions to experiment and apply possible techniques by my own, and helped me out when I faced with difficulties during experiments. The whole process is not easy considering that I was the first time with this area. Thanks for their help and support, I came across all the problems and completed the thesis.

I am also grateful to my family for their continuous love and support. They gave me trust and patience during my years of study. Without their believe, I would not have opportunities to discover and pursue my interests.

Furthermore, I want to thank my friends and fellow students for their participation with my thesis. All the conversations and discussions we had inspired and motivated ideas of the thesis. I am also appreciated for their help for solving issues and difficulties on both technical and theoretical sides.

Finally, I express my sincere gratitude to all the people who gave me help during my thesis. It is great to study here and my research skills and computer science knowledge have been enhanced all these years.

Contents

Chapter 1

Introduction

1.1 Motivation

Named entity recognition (NER) is one of the focus areas of Natural Language Processing (NLP) problem in recent years. The task is important for various applications, such as information extraction, syntactic analysis, machine translation and questions-answering systems. General named entity recognition is to detect names of persons, organizations or locations in texts and classify them into pre-defined categories. The approaches include dictionary-based methods, rule-based methods and statistic methods, such as Conditional Random Fields (CRF), all of which have achieved good performance for NER (He and Kayaalp, 2008; Abacha and Zweigenbaum, 2011b; Abacha and Zweigenbaum, 2011a). A separate class of NER is domain specific entity recognition problems. Medical entity recognition (MER) is one of them. The task aims to extract medical terms, such as names of drugs, genes and proteins, or diseases from medical corpus and clinic text, which is important for providing valuable information from the yearly increase of published medical literatures (Aman et al., 2014).

Apart from the common techniques, medical entity recognition methods commonly use domain-specific techniques. Medical sources such as medical dictionaries and ontologies like Unified Medical Language System (UMLS) are available for MER (Aronson, 2001). The use of them provides a preliminary baseline for medical entity recognition, and could bring positive effects when its combined with machine learning approaches. Some tools such as MetaMap have been developed to parse medical text and automatically extract medical entities by mapping words to the UMLS ontology, and the obtained results combined with statistic methods achieved the best performance (Abacha and Zweigenbaum, 2011c).

Most of studies aim to extract medical entities from electronic medical records or

medical reports, while as the rapid development of internet, there are more people using social networks for sharing information, like Twitter or Facebook. The information they provide online, such as their personal experiences with diseases, or sides effects they might have after taking some medicines, shows increasing importance for both patients and professional groups. The characteristics of medical data pose challenges for entity extraction, such as non-uniform naming rules, polysemy and synonymy. Moreover, unlike structured medical records and reports, text from online forums increases more difficulties because it is informal and unstructured user narratives. The non-standard use of language makes the text more ambiguous.

To address the difficulties, we propose an approach to extract medical entities from user-generated forum text. In this paper, we firstly perform an unsupervised UMLS lookup method and then a supervised conditional random fields model to further improve the performance. The dataset used in this paper is from Facebook GIST (Gastrointestinal Stromal Tumor) group[1]. Instead of using domain knowledge tools like MetaMap, we conduct the UMLS lookup by constructing a database loaded with UMLS Metathesaurus data to improve flexibility and realize custom selection of UMLS semantic types. Because some medical terms have one and more words, thus we also make use of N-gram noun phrase chunking on the text to detect multi-word terms. Furthermore, in order to improve the performance, we utilize unsupervised learning with word embeddings generated by the Word2vec tool and K-means clustering using Wikipedia data, and finally combine the results with a CRF sequence labeling model. The pipeline consists of preprocessing, tokenization and part-of-speech tagging, noun phrase chunking, UMLS database lookup, word embedding and clustering, and sequence labeling with features derived from the previous steps.

1.2 Research Questions

In this paper, we aim to improve the performance over the unigram UMLS lookup baseline, and propose an approach for medical entity recognition from patient forum data. Thus the questions are: (1) Are POS tagging and N-gram chunking be useful to improve the performance over the unigram UMLS lookup baseline? (2) For the entities that cannot be found in UMLS, and the ones that are wrongly mapped to UMLS, can a sequence labeling model be used to address the issues? (3) Are word embedding and clustering be useful to improve the results?

We set the result of the unigram UMLS lookup as our baseline for the UMLS lookup results evaluation. We also compare the performance of unsupervised UMLS lookup method with the supervised CRF method. In the supervised CRF experiments, we

[1] https://www.facebook.com/groups/gistsupport/

study the features used for sequence labeling, including word features, context features, POS features, an N-gram noun phrase feature, an UMLS look-up feature, and an unsupervised clustering feature generated by word embeddings trained on Wikipedia corpus. We also evaluate the contribution power of each feature to improve the performance of the sequence labeling model.

1.3 Thesis Structure

In Chapter 1 *Introduction*, we do introduction of the thesis, and list the research questions that we want to solve during the experiments.

In Chapter 2 *Background and Related work*, we explore the background knowledge of medical entity recognition, and give an explanation of related knowledge of the methods used in our approach, including the UMLS ontology, word embedding techniques and models, K-means clustering, and conditional random fields used for sequence labeling applications. For the related work, we introduce various existing methodologies, such as rule-based, statistic-based, and hybrid approaches, used in other papers to perform medical entity recognition from both medical literatures and user-generated content. We also compare the difference of these methods, and summarize the unique points of our proposed approach.

In Chapter 3 *Methodology*, we dive into details of each step performed in our approach and the way we incorporate them. We also describe the datasets and features that we use in this paper for training and testing models.

In Chapter 4 *Results*, we give the description of the evaluation method and present the results of the unsupervised UMLS lookup and supervised CRF methods. We also compare and discuss the results, and analyze them with error analysis.

In Chapter 5 *Conclusion and Future Work*, we draw conclusions of the proposed approach and shed light on suggestions for future work.

Chapter 2

Background and Related Work

2.1 Medical Entity Recognition

2.1.1 Introduction

With the development of medical information technology, a large number of electronic health records are available. This data is not only used for clinical decision support systems, but also for medical studies. Medical Entity Recognition (MER) is a sub-domain of Named Entity Recognition (NER) in Information Extraction (IE), which aims to transfer unstructured information in medical text into structured information, and these extracted medical instances can be directly utilized by structured clinical systems. Generally, medical entity recognition consists of two parts: (i) detecting medical entities in the text and (ii) determining their categories (Abacha and Zweigenbaum, 2011b). MER is the required first step of extracting the implicit semantic relations between medical entities, therefore the recognition efficiency is important for various applications, like automatic knowledge extraction systems. Nowadays, medical entity recognition is not limited to extract information from medical and clinical reports, but also from medical social media data. Patient social media provides a new source for information exchange, such as personal experiences of diseases, medical conditions, or patient discussions. The web social medial tools are widely used for medical-related information, and becoming more and more attractive for research interest (Denecke and Nejdl, 2009).

Some studies have applied MER techniques to extract medical information from tweets and social medial postings (Liu and Chen, 2015; Jenhani, Gouider, and Said, 2016). However, the general methods, like using MetaMap to directly extract medical terms in forum data (Tu et al., 2016), have poor performance. The nature of user-generated content poses a big challenge for this task, and the unstructured data

without labeling also increase the difficulty in applying machine learning based approaches. There has been limited progress regarding to this, and the performance of MER for social media still has much space for improvement.

2.1.2 Challenges

Medical entity recognition has its unique characteristics, which make the problem become more complicated than the normal NER tasks.

- The yearly increase amount of medical literatures and vocabularies pose a challenge that medical dictionaries and training datasets is difficult to maintain and provide sufficient information.

- Non-standard naming rules. Some medicines are known for their brand names other than scientific names, for example, "Imatinib" is generally known as its brand name "Gleevec". MER tools trained on one medical dataset could have poor performance on the other one.

- Word-formation is complex. Most of the medical terms have different spellings and expressions. In addition, medical terms usually have one and more words, including capital and lower-case letters, numbers or other kinds of symbols. The complexity of spellings and expressions makes medical entities easier to confuse.

- Polysemy and synonymy. On one hand, words could have different meanings based on context. On the other hand, a medical concept could also be expressed as different terms, including abbreviations. This enhances the difficulty of word ambiguity that MER systems are hard to distinguish them.

As for medical entity recognition from patient social media, the problem is becoming more challenging due to the non-standard use of language.

- Less grammatical language. User narratives always do not use canonical grammar rules, and are expressed more concise. The abundant word abbreviations or incomplete phrases are used in text, for example, "b4" means "before", "whipple" could mean "whipple procedure". The informal use of language brings more difficulties in detecting entities and solving ambiguity.

- Lack of context information. Social media like Twitter or Facebook postings are terse content, and unnaturally length of sentences without punctuation could

cause confusing expressions, thus the context information is not enough for MER systems to identify medical entities effectively.

2.2 Methodological Background

To address the issues and challenges mentioned above, we proposed our approach in this paper. The related methodology knowledge used in the approach are introduced in this subsection.

2.2.1 Unified Medical Language System

Unified Medical Language System (UMLS) is an integrating ontology in the medical domain developed by the National Library of Medicine (NLM). The NLM[1] is the largest biomedical library, which maintains and provides electronic information services for scientists, health professions and the members of public. The UMLS[2] is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems. It can be used to improve and develop applications, such as retrieval systems of medical information. The use of the UMLS also makes it possible for finding relationship for medical concepts, such as drug names and medical terms. Some other applications include public health statistic reporting, or linking health information across different computer system (Medicine (US), 2009).

The UMLS knowledge resources are the Metathesaurus, the Semantic Network, and the SPECIALIST Lexicon.

- ***Metathesaurus*** The Metathesaurus[3] is a huge vocabulary database that includes information about biomedical and health related concepts, their various names, and the relationships among them. It is built from various sources, such as public health statistics, biomedical literature, and health service research. The scope of the Metathesaurus is determined by the combination of its sources vocabularies. All concepts which appear in source vocabularies are recorded in the Metathesaurus, and the meaning of each concept is defined by its source.

[1] https://www.nlm.nih.gov/about/
[2] https://www.nlm.nih.gov/research/umls/quickstart.html
[3] https://www.ncbi.nlm.nih.gov/books/NBK9684/

The Metathesaurus is organized by concepts, which are associated to alternative names, synonyms concepts and the relationship between different concepts. Each concept has a Concept Unique Identifier (CUI), which links the Metathesaurus to the other UMLS knowledge sources, the Semantic Network and the SPECIALIST Lexicon.

- ***Semantic Network*** The Semantic Network[4] contains broad categories, or Semantic Types, and essential relationships, or semantic relations, of existing semantic types. The Semantic Network provides information about a consistent semantic types for all concepts in the Metathesaurus, and possible relationships between them. Each concept has at least one semantic types. The semantic types are the nodes in the Network, and the relationship of them are linked to one another. For now, the Semantic Network has 135 semantic types and 54 relationships. The major groups of semantic types are for organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas.

- ***SPECIALIST Lexicon and Lexical Tools*** The SPECIALIST Lexicon[5] is developed for SPECIALIST Natural Language Processing System. It provides lexical information for general English biomedical terms. The syntactic, morphological, and orthographic information is also provided by the system for each lexicon entry for each word or terms. The Lexical Tools are designed for normalizing words and terms in order to address the issues of lexical variations.

Some medical NLP tools, such as MetaMap, cTAKES and YTEX are developed based on the UMLS Knowledge sources. MetaMap (Aronson, 2001) is a highly configurable program, which can map biomedical text to the UMLS and find the Metathesaurus concepts. MetaMap conducts lexical or syntactic analysis, such as word sense disambiguation for input text. It arises in the context of an effort to improve biomedical text retrieval, specifically the retrieval of MEDLINE/PubMed (Aronson and Lang, 2010). It also provided a link between the text of biomedical literature and the knowledge, including synonymy relationships, embedded in the Metathesaurus. Another such tool is Clinical Text Analysis and Knowledge Extraction System (cTAKES) (Savova et al., 2010), which is an a comprehensive modular system for medical information extraction from electronic clinical text. It uses a subset of the UMLS as a dictionary, and enriches the dictionary with synonyms from the UMLS and Mayo-maintained terms. Yale cTAKES (YTEX) (Garla et al., 2011) extends the application of cTAKES pipelines. It enhances the performance by word sense disambiguation and appending unclassified lexical variants of clinical concepts to the UMLS dictionary.

[4]https://www.ncbi.nlm.nih.gov/books/NBK9679/
[5]https://www.ncbi.nlm.nih.gov/books/NBK9680/

The UMLS is a multi-purpose resource, and the program named MetamorphoSys makes it possible for an effectively use of the knowledge sources for specific applications. The MetamorphoSys[6] in the software tool which enables public users to install one or more UMLS Knowledge Sources, or create customized Metathesaurus subsets. It also provides output formats for different computer operating systems and software. In this paper, we used this application to download the UMLS Metathesaurus, and created a database with MySQL to find possible semantic types for each term entry.

2.2.2 Word Embeddings

2.2.2.1 One-hot Representation

One-hot representation is an intuitive way to represent a word as a vector. It represents a word as a one-hot vector in which only one element is "1" and the rest of elements are "0". The place of "1" in the vector stands for the corresponding word. The dimension of the vector is the vocabulary size, which makes this kind of word representation sparse and high-dimensional. One-hot representation simply distinguishes one word in the vocabulary from another, but does not capture semantic similarity and relationships between words.

2.2.2.2 Distributed Representation

Another word representation method is called distributed representation, which was originally proposed by Hinton in 1986 (Hinton, 1986). Distributed Representation represents a word as a low dimensional real number vector, usually ranging from 50 to 400 dimensions. Each dimension of a vector represents the context of the word, and words that occur in a similar context have similar distributed vectors. The advantage of this method is that the similarity between words can be calculated with spatial distance or cosine similarity. Words that are close to each other in the vector space thus have similar meanings. The low-dimension vectors also enable computationally efficiency.

Word embedding is realized based on the distributed representation. The word embedding techniques automatically learn low-dimensional vectors and generate mapping functions where word or phrases in the text are mapped to vectors of real-values numbers. Methods for generating mapping functions include neural

[6]https://www.ncbi.nlm.nih.gov/books/NBK9683/

networks (Mikolov et al., 2013a) and matrix factorization (Pennington, Socher, and Manning, 2014).

2.2.2.3 Word2vec

Word2vec is a state-of-art tool developed by Thomas Mikolov in 2013 for word embeddings (Mikolov et al., 2013a). The models used in the tool are based on the distributed representations of words. Word2vec simplifies input text as vector representations in a low dimensional vector space, and the similarity in the vector space can be expressed as semantic similarity of words or phrases. This property has two important characteristics. First, it shows semantic similarity relations. For example, the nearest neighbors of word "red" are most likely to be "white" or other words representing colors. Second, it shows linear translation relations. Once words have been mapped to the vector space, it is possible to use vector addition to find words which have analogical semantics. For example, vec("Paris") - vec("France") + vec("Italy") is closer to vec("Rome") than to any other word vectors. The output word vectors also have subsequent applications in NLP, such as clustering.

The implementation of Word2vec can achieve a better performance and improve computational efficiency compared to other algorithms (Mikolov et al., 2013a). For example, the skip-gram model (Mikolov et al., 2013b) can efficiently learn high-quality word vectors from large amounts of unstructured text data, and train on more than 100 billion words in one day with an optimized single-machine implementation. There are two architectures used in the Word2vec tool, the continuous bag-of-words (CBOW) and the above-mentioned skip-gram models, for computing word embeddings. The following is the description of these two models:

- **Continuous Bag-of-words Model**

Figure 2.1 shows the structure of the CBOW model. It has three layers, input, projection and output, where the hidden layer is removed and projection layer is shared for all words. The training object of the CBOW model is to predict the target word based on the surrounding words in a sentence,

$$P(w_t|w_{t-k}, w_{t-(k-1)} \ldots, w_{t-1}, w_{t+1}, w_{t+2} \ldots, w_{t+k})$$

More precisely, the operation from the input layer to the hidden layer is actually the addition of the context vector, and then projected to calculate the probability of the current word occurrence. The model is continuous bag-of-words as the orders of words in the history has no influence on projection.

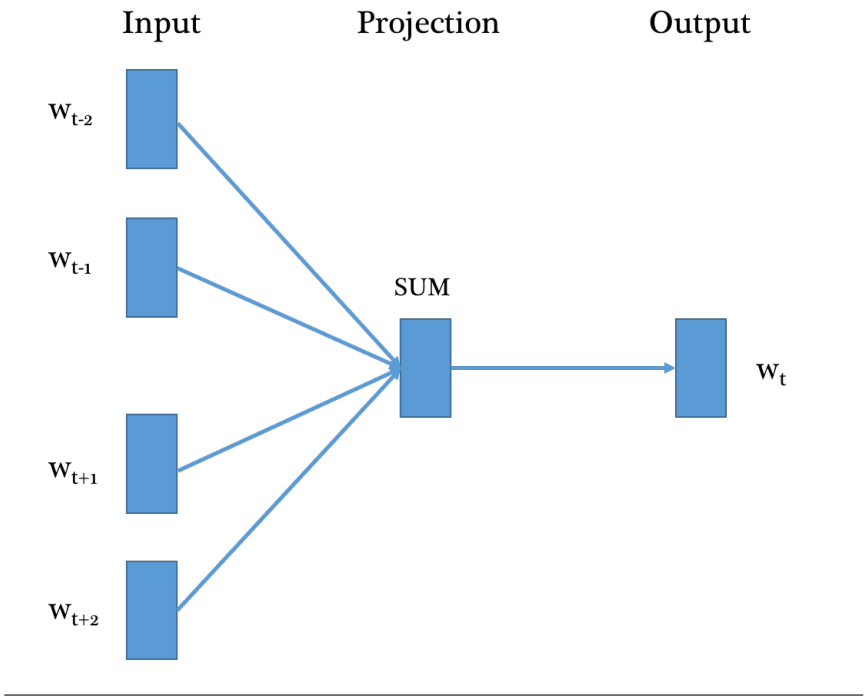

FIGURE 2.1: Continuous Bag-of-words Model

• **The Skip-gram Model**

Figure 2.2 is the structure of the skip-gram model, which has a similar structure with CBOW, while the prediction is "inversed". It uses the current word to predict the surrounding words in a sentence or document. With a given sequence of training words, the skip-gram model sums the log probability of the surrounding words on the left and the right of the current word. The objective of the model is to maximize the average log probability (Mikolov et al., 2013a):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq k\leq c, k\neq 0}\log p(w_{t+k} \mid w_t)$$

where T is the size of training words, and c is the size of the training context of the current word w_t. The larger training corpus can result higher accuracy because of more training examples.

According to evaluation of both models (Mikolov et al., 2013b), the skip-gram model achieves a better performance over the other due to the fact it needs to train on more words to make predictions. Therefore, the skip-gram is more sensitive to infrequent words or phrases, but CBOW is several times faster and more suitable for a large corpus. All in all, the overall performance of the Word2vec tool depends on the choice of the model architecture, the dimension of the vectors, the setting of the subsampling rate, and the size of the training window (Mikolov et al., 2013a).

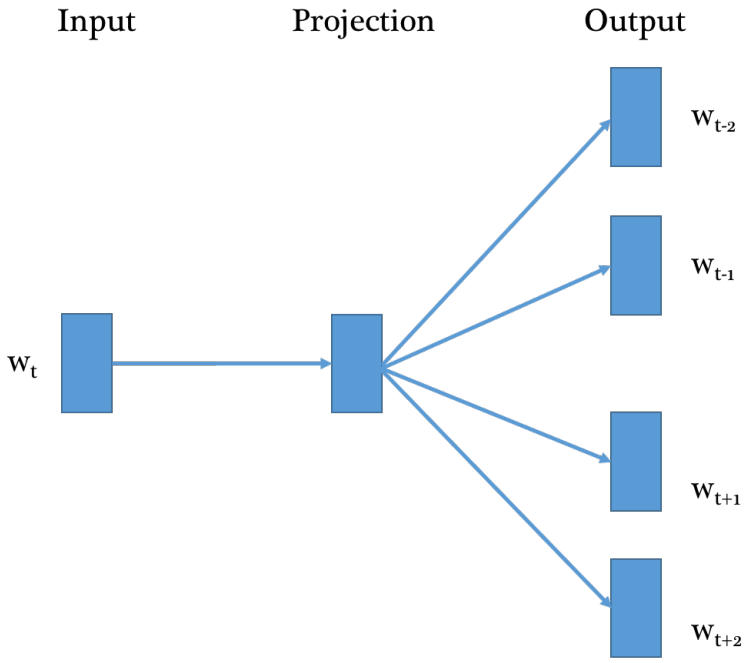

FIGURE 2.2: The Skip-gram Model

2.2.3 K-means Clustering

Clustering is the process to partition or group a collection of data into a number of clusters. These clusters share similar features. K-means clustering (MacQueen, 1967) is one of the important unsupervised clustering algorithms when no labeling data is available. The goal of k-means clustering is to classify a given dataset into predefined the number of k clusters. The main idea is first to decide k centroids (clusters), and second to cluster the points to the nearest k centroids.

Generally, k-means clustering algorithm uses iterative optimization to get the final result. The standard k-means algorithm begins with randomly selection of k centroids, and then iterates with two following steps:

(1) Assignment step

Each centroid means a single cluster. For every point in the dataset, it is assigned to the nearest centroid based on the formula, which is defined below:

$$c^i = \arg\min_j d(x_i, \mu_j)$$

where $d(x_i, \mu_j)$ is the distance between the point x_i and the centroid μ_j, and c^i is the cluster assigned to the point x_i. Usually, the squared Euclidean distance $d(x_i, \mu_j) = \|x_i - \mu_j\|^2$ is used as the distance measurement.

(2) Centroids update step

After the assignment step is done, the centroids are recalculated by setting the position of each cluster to the mean of all data points attributed to that centroid clustering.

$$\mu_j^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{x_j \in C_i^{(t)}} x_j, \forall i$$

where C_i is the set of the data points in the cluster and $\mu_j^{(t+1)}$ is the newly updated centroid.

The algorithm iterates between the steps until convergence meaning that when no changes of the assignments or reaching the maximum number of iterations. Finally, the algorithm aims to minimized an objective function, a sum of squares of errors function defined as:

$$\arg\min_{\mathbf{c}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathbf{c_i}} \|\mathbf{x} - \mu_{\mathbf{i}}\|^2$$

Although k-means clustering algorithm is proved to converge at last, it does not guarantee to produce a global optimal result. The number of cluster is determined by pre-selected k, while there are no methods to determine the optimal value of k. Besides, the performance is also sensitive to the random selection of starting centroids. Therefore, it is worth to run k-means clustering algorithm multiple times to find the optimal k and a relatively better performance (Likas, Vlassis, and Verbeek, 2003).

2.2.4 Conditional Random Fields

Conditional random fields (CRF) is one of the most popular algorithms in Natural Language Processing in recent years, commonly used in syntax analysis, named entity recognition, and part of speech tagging. CRF (Lafferty, McCallum, and Pereira, 2001) uses a probabilistic model for sequence labeling and segmentation data. Comparing to other probabilistic models, such as Hidden Markov models (HMMs) and Maximum Entropy Markov models (MEMMs), CRF attains all the advantages and solve the label bias problems existed in MEMMS and Markov directed graphical models (Wallach, 2004). This makes CRF perform purely based on probability distributions.

CRF is viewed as an undirected probabilistic graphical model. Given a random

variable X to be labeled and a random variable Y is over corresponding label sequences, where X and Y are jointly distributed. The target of the model is to compute the conditional probability distribution $P(Y|X)$. To calculate the conditional probability of a label sequence $Y = [y_1, y_2, \ldots, y_n]$, given an observation sequence $X = [x_1, x_2, \ldots, x_m]$, the form of the joint distribution defined by (Lafferty, McCallum, and Pereira, 2001) as:

$$P(Y|X) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i)\right)$$

in which $Z(x)$ is a normalization factor, defined as:

$$Z(x) = \sum_{y} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i)\right)$$

where t_k and s_l are transition feature functions, and all features are real-values as 1 or 0. λ_k and μ_l are the corresponding parameters to be estimated for feature functions during training. Noting that the sum is done on all possible output sequences.

CRF is a single exponential model used for the joint probability over the entire sequences and given labels, which has a benefit that the weight of different features is trade-off against to each other. The parameter learning can be trained by maximum likelihood estimation or limited-memory BFGS (Wallach, 2002) to maximize the log-likelihood of training data.

Feature extraction is important for a CRF model, which determines feature functions generated in the model and has an effect on the final performance. The choice of features is not the more the better. Redundant features could influence the efficiency of prediction and result "over-fitting" in the performance. While insufficient features could provide limited information for the model and lower the precision and recall. Therefore, it is considerable to select features that have higher correlation characteristics of input datasets.

In general, CRF is a robust method and has various applications for NLP tasks in different domains. In this paper, we used this method to perform the sequence labeling tasks.

2.3 Related Work

Various researchers have studied different methods for named entity recognition in medical domain, which can be simply divided into three approaches: unsupervised, supervised and hybrid approaches. Unsupervised methods for MER usually are rule-based, which makes use of linguistic rules or patterns, and dictionary-based lookup to recognize medical related entities. On the other hand, supervised methods use classifiers or sequence labeling techniques with the help of various features to train a model to make classification or label prediction of entities. The performance of supervised methods is better than unsupervised methods, while the latter has an advantage that no training data is required. Hybrid methods combining the advantages of unsupervised and supervised approaches, have become popular in recent year (Abacha and Zweigenbaum, 2011c; Jenhani, Gouider, and Said, 2016).

2.3.1 Unsupervised Methods

Cohen (Cohen, 2005) constructed a dictionary-based gene and protein named entity recognition and normalization system. The dictionary was automatically generated from five databases, including MGI, Saccharomyces, UniProt, LocusLink, and the Entrez Gene database. As the development of medical ontology such as the UMLS and the associated applications like MetaMap, some studies utilize such resources to automatically recognize medical entities from biomedical text. Kang et al. (Kang et al., 2014) created a knowledge-base, which was filled the data from the UMLS ontology to extract adverse drug events from biomedical abstracts. Abacha and Zweigenbaum (Abacha and Zweigenbaum, 2011b) developed an annotation approach based on linguistic patterns and domain knowledge. They performed medical entity recognition by the use of MetaMap, and enhanced the performance of MetaMap by using an external sentence segmenter and noun phrase chunker.

Apart from rule-based and dictionary-based methods, another kind of unsupervised method involves bootstrapping-like techniques. Zhang and Elhadad (Zhang and Elhadad, 2013) introduced a stepwise unsupervised method to biomedical named-entity recognition. There are three main steps in this approach: seed term collection from the UMLS Metathesaurus; entity boundary detection by chucking noun phrases and followed by a filter to avoid non-entity noun phrases; entity classification by feeding all the candidate entities into a classifier and calculating their similarity with signature vectors to predict their semantic types.

2.3.2 Supervised Methods

Many papers make contributions to the use of supervised machine learning methods for medical entity recognition. Bodnari et al. (Bodnari et al., 2013) developed a supervised conditional random fields model which was combined with various features to predict disorder named entities from electronic medical records. The CRF model used a rich feature set and external knowledge sources from UMLS and Wikipedia. This paper suggested that including brown word clustering features may improve recall by detecting out-of-vocabulary words. He and Kayaalp (He and Kayaalp, 2008) showed a statistical machine learning methods which untilzed UMLS semantic types features from MetaMap, SemRep, and ABGene, and a CRF model to extract biomedical entities on GENIA corpus. The features used in the CRF model included UMLS semantic types, part-of-speech, orthography, and gene name information. The results showed that both MetaMap semantic features and orthographical features can improve the overall performance. However, the limitations of the available open source software packages were prominent.

Recently, features derived from unsupervised learning methods have been used to further improve the performance of machine learning models. Sekppstedt (Skeppstedt, 2014) proposed experiments of the usefulness of features extracted from unsupervised method by performing named entity recognition within one clinical subdomain and adapting the model to a new clinical subdomain. Features from unsupervised machine-learning methods can be generated by clustering techniques which transfer semantic representation of the word space model into features. Tang et al. (Tang et al., 2014) compared three different types of word representation features for biomedical named entity recognition, including clustering-based, distributional representation and word embeddings. They incorporated these three unsupervised features with a CRF model. This paper used the Word2vec tool to generate word embedding features and proved all the unsupervised features can improve performance.

2.3.3 Hybrid Methods

Abacha and Zweigenbaum (Abacha and Zweigenbaum, 2011c) presented and compared semantic and statistical methods based on domain-knowledge and machine learning techniques. In this paper, they studied three different approaches. First was the sematic method relying on MetaMap. Second was chunker-based noun phrase extraction and SVM classification to obtain the maximal number of right noun phrases and filter out the irrelevant ones. The third step was to use a CRF model. By comparing the three different methods, they reached the conclusion that

the hybrid method which combined the CRF model with semantic features obtained from a domain-knowledge based method using MetaMap had the best performance. Zhang et al. (Zhang et al., 2016) developed a system for the Chemical Entity Mention Recognition in Patents. They applied two additional features above the baseline: domain knowledge features from chemical/drug dictionaries, chemical patterns and semantic types from UMLS; word representation features generated by unsupervised brown clustering. The results showed that each of the additional features improved the performance of the CRF-based and SSVMs based systems.

2.3.4 Medical Entity Recognition from User-generated Content

Most of papers listed above used annotated datasets or clinic text as input datasets, which are formal and structured, while limited research is regarding to the medical entity recognition from user-generated content. Sondhi et al. (Sondhi et al., 2010) developed a system to extract two related types of sentences, medical problem and medical treatment, from medical forum data. They manually labeled forum data and trained the data with support vector machines and conditional random fields models. The overall accuracy of their system was 75%. Nikfarjam et al. (Nikfarjam et al., 2015) introduced an ADRMine system to extract mentions of adverse drug reactions (ADRs) from Twitter and DailyStrength (DS), which is a health-related network, based on a CRF method. They combined semantic features into the model, which were generated by the Word2vec tool and K-means clustering. Their word2vec model was trained on additional user reviews from DS. The paper showed that the CRF model with word embedding clustering features can improve the performance over MetaMap baselines, with F-measure of 82% for DS and 72% for twitter datasets.

The informal user-generated data enhances difficulties in processing data as well as recognizing medical entities because of mistake spellings and word abbreviations. To address the difficulties and improve the performance, we propose a novel approach in this paper with the insights given by the pervious work to extract medical entities from patient forum data. We firstly perform an unsupervised UMLS lookup method with both unigram tokenization and n-gram noun phrase chunking. The existing tools like MetaMap are commonly used to find corresponding UMLS semantic types of input terms. However, MetaMap performs poorly on social media data (Tu et al., 2016), which is mainly because it is designed for processing medical terminologies and has limitations in solving the problems caused by the forum content. Therefore, we conduct our UMLS lookup method by constructing a database with information from the UMLS Metathesaurus. This can enhance flexibility in

processing data and performance of medical entity recognition. We also utilize a supervised CRF model combined with unsupervised word embedding clustering features to further improve the performance. Unlike Nikfajam et al. (Nikfarjam et al., 2015), who trained word embeddings with additional user posts from the same social network for adverse drug reactions, we generate word embeddings with unannotated medical-related Wikipedia articles, which can have more general applications. Moreover, apart from the common word, context and POS tag features used for the CRF model, we also add two additionally external features collected from the UMLS lookup results and noun phrase chunking to improve the CRF performance. The detailed description of each pipeline of our approach is illustrated in the following section.

Chapter 3

Methodology

Our medical entity extraction approach is aiming to extract medical entities from a given patient forum data. In this paper, we performed both unsupervised and supervised methods for this task. We firstly used unsupervised unigram and N-gram UMLS database lookup, and then combined them with word embedding clustering as input features for supervised CRF sequence labeling to further improve the performance. The steps are described in the following subsections. Our work is implemented in Python 2.7, and MySQL is applied to create a database for loading UMLS Metathesaurus data.

3.1 Preprocessing

The first step is to preprocess forum text. Forum text is conversational, ungrammatical and informal, and it often contains abbreviations, external links or non-standard characters. Before tokenization, we firstly have to clean it. We removed punctuation in sentences, URLs and non-alphanumeric characters from text by using regular expressions, and we kept hyphens within a word, such as “insulin-like” remained the same instead of “insulinlike”.

3.2 Tokenization and Part-of-speech Tagging

Tokenization and part-of-speech tagging can be useful to recognize medical words by their linguistic characteristics, for example, most of medical terms are nouns and adjectives, thus we can use part-of-speech to remove words which are in other classes. We used the NLTK toolkit to do tokenization and POS tagging.

3.3 N-gram Noun Phrase Chunking

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. Medical terminology, such as CT scan, red blood cell or insulin-like growth factor, often contains more than one words, thus in order to detect these medical terms, we perform bigram and trigram noun phrase chunking. The tool used to collect noun phrases is NLTK. It can recognize chunks based on the part-of-speech tags and chunk grammars, which consist of rules that define how sentences would be chunked. We set up two kinds of chunk grammars to get chunking sequences. One is the common noun phrase combination, *adjectives + noun(s) + noun(s)(optional)* and in case that some adjectives could be wrongly tagged as nouns, we also add the other form *noun(s) + noun(s) + noun(s)(optional)*. All sequences are up to three words. Table 3.1 lists the examples of each chunking sequence.

Chunk grammar	N-gram	Example
<NN.*><NN.*>	Bigram	B12/NNP injection/NN
<NN.*><NN.*><NN.*>	Trigram	Small/NNP bowel/NN resection/NN
<JJ><NN.*>	Bigram	intestinal/JJ sarcomas/NN
<JJ><NN.*> <NN.*>	Trigram	insulin-like/JJ growth/NN factor/NN

TABLE 3.1: Chunk grammars and examples

3.4 Unsupervised UMLS Lookup Method

There are some advanced tools developed to parse medical text relying on the UMLS, such as MetaMap or cTAKES introduced in the last section. Some papers used MetaMap to map words and noun phrases in raw texts to get UMLS semantic types according to their matching score. While it is a state-of-art tool, it does not completely solve the medical entity recognition for forum data. First, the UMLS is served as huge medical ontology, and contains more than 100 semantic type, but not all of them are related to terms that we care about, like the semantic type of Animal. Second, forum data contains many abbreviations and slangs, which might be wrongly detected as medical entities, or medical entities might be missed because of misspellings by using such tools.

Thus to improve flexibility and select semantic types we are most interested, we used the UMLS Metathesaurus MySQL load scripts to load the Metathesaurus data into a database for UMLS semantic type lookup. The version used is *UMLS 2016AB Active Release*. We created a Concept Unique Identifier (CUI) as index for different tables to

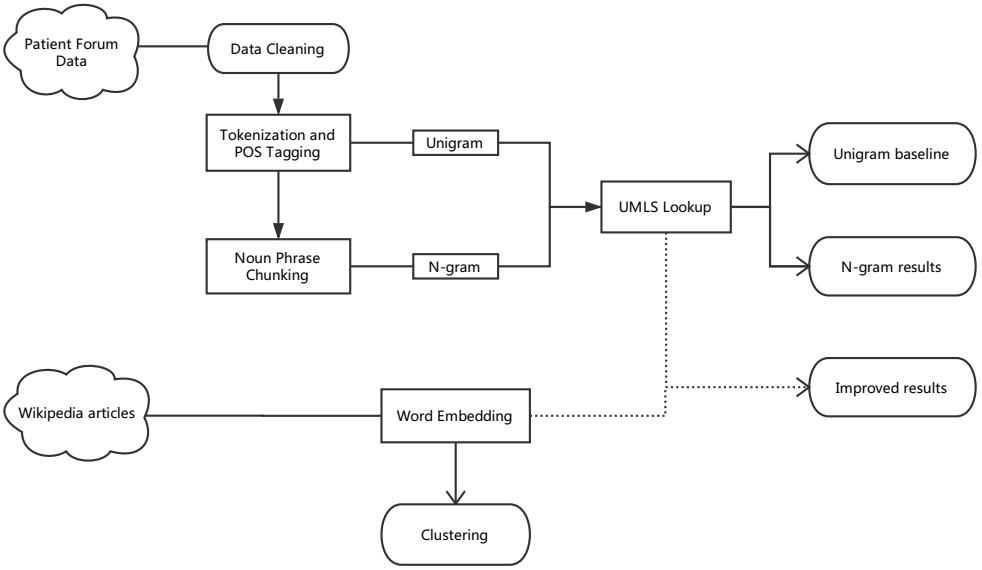

FIGURE 3.1: Unsupervised UMLS Lookup Framework

do queries more efficiently. The overview framework of UMLS lookup procedures is illustrated in the Figure 3.1.

There are total 135 semantic types in UMLS, and we select 20 of them we concern about, which are listed in Table 3.2. Words and noun phrases extracted from tokenization and noun phrase chunking are mapped to the database, which then returns their semantic types. One word may have multiple same or different semantic types according to its CUI, and the most frequent semantic type is chosen as the category. If two or more semantic types have the same frequency, we choose the first one appearing in the predefined 20 categories as the semantic type. If a word or a noun phase can be found with a corresponding semantic type in the UMLS, it is counted as a medical entity by the system. Besides, a list of English stop words[1] is used prior to the lookup procedure to filter out irrelevant common words. At the end of the UMLS lookup, we output and stored UMLS lookup results into a UMLS entities dictionary to improve the lookup efficiency.

3.5 Word Embedding and Clustering

It is worth mentioning that not all medical terms could be found in the UMLS database, and on the other hand, some short words or colloquial word expressions

[1]http://xpo6.com/list-of-english-stop-words/

Semantic Types	
Biomedical or Dental Material	Disease or Syndrome
Cell or Molecular Dysfunction	Diagnostic Procedure
Gene or Genome	Medical Device
Chemical	Amino Acid, Peptide, or Protein
Organic Chemical	Pharmacologic Substance
Food	Sign or Symptom
Neoplastic Process	Pathologic Function
Body Substance	Biologically Active Substance
Tissue	Body Location or Region
Body Part, Organ, or Organ Component	Therapeutic or Preventive Procedure

TABLE 3.2: Selected Semantic Types

could also be wrongly detected as medical entities during UMLS lookup. Thus to avoid the situations, we performed word embeddings for the forum data. Word embeddings recently have been performed to various NLP tasks. Word embedding techniques map words or noun phrases to low-dimensional vectors of real numbers, and then similarity or clustering methods could be used based on the vectors.

The use of word embeddings could group similar words according to the context. In this paper, we use the techniques to generate word vectors and further compute word clusters. In this way, some words or noun phrases that have been wrongly identified as medical entities can be filter out. The combination of clustering features and the later sequence labeling model can be used to detect unseen or rare occurrence words in the text and finally improve the overall performance.

3.5.1 Word2vec with Wikipedia Corpus

Word2vec is a group of related models based on neural networks that takes a corpus of text as input and outputs word embeddings. Each unique word in corpus will be mapped to a vector space, and words which share common context have similar vectors. Word2vec can either use continuous bag-of-word or skip-gram model to produce distributional representation of words. We perform word embeddings through Python Genism Word2vec package[2].

In this paper, we collected word embeddings using the skip-gram model, which is proved having a better performance for infrequent words. We trained the model with total 3930 Wikipedia pages from 20 medical-related Wikipedia categories and two additional pages. The articles are extracted with Wikipedia *Special:Export*[3] tool

[2]https://radimrehurek.com/gensim/models/word2vec.html
[3]https://en.wikipedia.org/wiki/Special:Export

and parsed with the *Wikiextractor* python script[4]. Before training, we firstly preprocessed and tokenized the Wikipedia corpus, and then lowercased each token for normalization. We generated word embeddings with 100 dimensions and the minimal word frequency is set 6.

Articles	**No.of tokens**	**Vocabulary size**
3930	2360050	18718

TABLE 3.3: Wikipedia Corpus Dataset

Moreover, the Word2vec tool can generate a vocabulary set of input corpus, and we find this can be used to improve the precision of the UMLS lookup method. Wikipedia corpus can further filter out low frequent rare words or word abbreviations which are not shown in Wikipedia corpus but have been wrongly detected as medical entities during the UMLS lookup process. For example, word abbreviation "b4", meaning "before" in colloquial way, which is wrongly assigned to "Amino Acid, Peptide, or Protein", can be removed during this step and thus enhance the precision of UMLS lookup results. Wikipedia corpus and Wikipedia category information are presented in Table 3.3 and Table 3.4 respectively.

Wikipedia Categories	
Genes on human chromosome 1	Surgical oncology
Cancer treatments	Experimental cancer drugs
Oncology	Antiemetics
Gastrointestinal cancer	Diseases of intestines
Gastrointestinal tract disorders	Antidiarrhoeals
Receptor tyrosine kinase inhibitors	5-HT3 antagonists
Tyrosine kinase receptors	Bones of the thorax
Sarcoma	Carboxylic acids
Rare cancers	Experimental cancer drugs
World Health Organization essential medicines	Diseases of oesophagus, stomach and duodenum
Page: Succinate dehydrogenase	Page: Darbepoetin alfa

TABLE 3.4: Selected Wikipedia Categories

3.5.2 K-means Clustering

K-means clustering can partition n points into different K clusters, where the distance between each point and its centroid is minimum. Because forum data is unlabeled, k-means clustering is used along with word embeddings got from Word2vec to produce word clusters, which are utilized as unsupervised features in the CRF model, to further improve the performance of sequence labeling. We computed

[4]https://github.com/attardi/wikiextractor

word clusters with different K values (the number of clusters: 100, 200, 300 and 400) by sklearn k-means tool[5].

3.6 Supervised learning with Conditional Random Fields

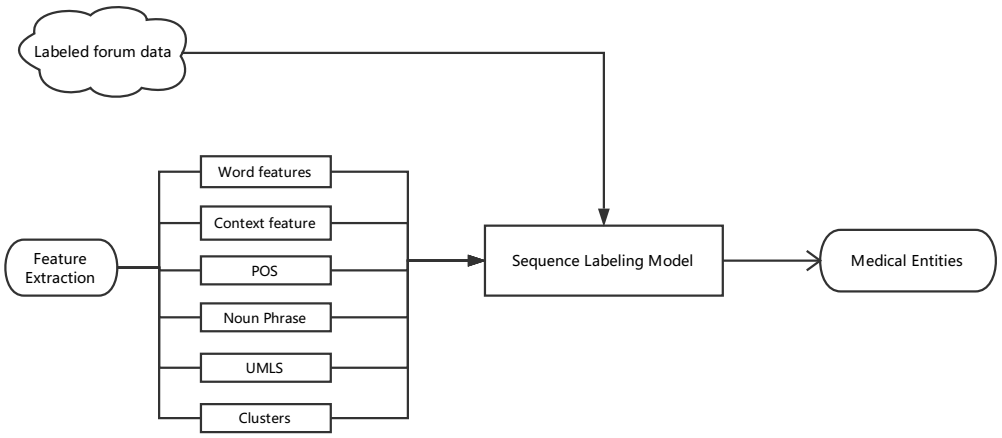

FIGURE 3.2: Supervised CRF Framework

In the previous sections, we performed the unsupervised UMLS lookup method, and we found there was still room for improvement. Therefore, we utilized a supervised Conditional Random Fields (CRF) model with various features in order to further improve performance. CRF is tested as an efficient machine learning model for sequence labeling tasks. The technique can classify input tokens based on feature sets and predict their labels. Here we use it incorporating with features derived from the former steps to improve the results of the UMLS lookup. In our experiments, the tool used is Python sklearn-crfsuite[6]. Figure 3.2 shows the framework of the CRF process.

3.6.1 Data Annotation

CRF is a supervised training model, thus we manually labeled forum data to train and evaluate the model. We have two raters, each of whom labeled data independently under the assumption that all labeled medical terms are medical entities, including signs or symptom, disease or drug names, body parts or organs and so on.

[5]http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[6]https://sklearn-crfsuite.readthedocs.io/en/latest/

Multi-word terms are also included. Then we calculated inter-rater agreement and made the ground truth. To perform sequence labeling, BIO-label schema (Siefkes, 2006) is used for labeling data. "B" means the current word is an entity or a beginning of an entity; "I" means the current word is the inside of a medical entity; "O" means words are the outside of entities. We labeled total 300 posts which were randomly selected from the raw text. 200 posts of them are for training and the remaining 100 posts are for testing. The inter-rater agreement of training and testing datasets is presented in Table 3.5, noting that the average of the percentage of agreement is 90.23%. The different entities respectively take up 6.76% and 12.59% of each rater's total number of entities.

	200 posts	**100 posts**	**Total**	**% of agreement**
Total entities of rater 1	681	488	1169	
Entities only by rater 1	45	34	79	6.76%
Total entities of rater 2	714	533	1247	
Entities only by rater 2	78	79	157	12.59%
Agreed entities	636	454	1090	**90.23%**

TABLE 3.5: Inter-rater Agreement

3.6.2 Feature Extraction

One benefit of CRF is that it makes it possible to combine any number of features, including the features of words themselves, and external features. For a CRF model, proper feature selection is critical because it determines feature functions and has direct effect on the performance. If feature set is too large, it could influence the training efficiency and cause "over fitting". While if feature set is small, the performance could be affected. Context features, POS features, orthographical features and syntactic features are commonly used as features in CRF models (Bodnari et al., 2013; Suárez-Paniagua, Segura-Bedmar, and Martínez, 2015; He and Kayaalp, 2008). Additionally, Bodnari et al. (Bodnari et al., 2013) used UMLS information from three sources (cTAKES, MetaMap, and directly search in UMLS) and Wikipedia category classification as input features for their CRF model. Suárez-Paniagua et al. (Suárez-Paniagua, Segura-Bedmar, and Martínez, 2015) applied unsupervised word embedding clustering to enhance the performance.

Based on previous studies and our observation of the characteristics of the input datasets, we therefore uniquely designed the feature set in this paper. In our approach, we used apart from the common POS features, lexical and syntactic features, we as well incorporated UMLS lookup results as a binary feature and unsupervised word embedding clustering features. For the context feature, we set window size of

1 according to the size of the training set, consisting of one previous and following tokens. The full description of each futures of a token is as follows:

- **Word feature** Word feature includes a token's characteristics: token is digit; token is upper case; token starts with uppercase; the lowercase form of a token. We also include a token's 3 and 2 suffixes. Table 3.6 is an example of word features.

Word	Is digit	Is supper	Is title	Lowercase	Word[-3:]	Word[-2:]
GLEEVEC	False	True	False	gleevec	VEC	EC

TABLE 3.6: Word Feature Example

- **POS tag** Part-of-speech of the current token, generated by the NLTK tool.

- **UMLS feature** A binary feature indicating that if the current token is in UMLS entities dictionary. If it is in the dictionary, this means the token has its corresponding semantic types during the UMLS lookup.

- **Noun phrase feature** We include the bigram forms of $w_{t-1}w_t$ and w_tw_{t+1} over the current token w_t. A "NP" label of the current token w_t if its bigram forms can be found in the UMLS dictionary, meaning it is a part of noun phrases. For example, both tokens "side" and "effects" have a "NP" feature because "side effects" is a noun phrase.

- **Clustering feature** The K-means cluster number of the current token. It also indicates if a token is in Wikipedia corpus. If no corresponding cluster number found for a token, then it returns "0" meaning the token is not in Wikipedia corpus.

- **Context feature** We define context feature with one preceding w_{t-1} and one following w_{t+1} words of the current token w_t. This is also along with the other features of w_{t-1} and w_{t+1}.

We also individually evaluate the contribution power of each feature, and compare the results of supervise learning with UMLS lookup results. To test the best values for our system, we perform different experiments with different K values on tuning dataset, which consists of another additional annotated 100 posts.

Chapter 4

Results

4.1 Dataset

We used discussion threads from the GIST (Gastrointestinal Stromal Tumor) patient support group[1] at Facebook to evaluate our methods. The forum provides a platform for GIST patients and families to share knowledge and support. The whole dataset contains 28927 posts and 974388 tokens.

	No.of posts	**No.of tokens**
GIST forum data	28927	974388

TABLE 4.1: Description of Patient Forum Data

To train and test the model, we used the manually labeled data described in the previous section. Moreover, we selected an additional 100 posts as a tuning set for optimizing the number of clusters k in k-means clustering (see Section 4.4). Table 4.2 shows the description of each dataset.

Dataset	**No.of posts**	**No.of tokens**
Training set	200	7435
Testing set	100	4851
Tuning set	100	3839

TABLE 4.2: Description of Datasets

4.2 Evaluation Method

We evaluate the performance of our approach using the standard measures of precision, recall, and F-measure, which are defined as follows:

[1]https://www.facebook.com/groups/gistsupport/

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

We evaluate the labels on the level of complete entities rather than on the word level. For example, a noun phrase "side effects" should be labeled as "B" and "I" respectively at the same time, while if either of them is labeled with other labels by the system, such as "side" is with "B" and "effects" is with "O", thus the entity being recognized as "side" instead of "side effects", we therefore regard it as a wrong entity.

4.3 Unsupervised UMLS lookup results

We implemented the UMLS lookup method mentioned in the last section. Table 4.3 shows the results of the UMLS lookup. We performed only unigram UMLS lookup as the baseline of the unsupervised UMLS lookup method. For the subsequent work, we added POS tags and noun phrase chunking to test their effectiveness and improve the performance. Additionally, we found that applying the Wikipedia corpus vocabulary collected via Word2vec can further enhance the overall performance. All the results of the UMLS lookup are unsupervised methods.

Configuration	Settings	Precision	Recall	F-measure
1	Unigram UMLS lookup	71.43%	70.18%	70.80%
2	1 + POS tags	77.21%	67.92%	72.27%
3	2 + NP chunking	79.42%	**82.21%**	80.79%
4	3 + Wikipedia corpus	**85.56%**	80.20%	**82.79%**

TABLE 4.3: Unsupervised UMLS Lookup Results

The results show that POS tags, noun phrase chunking and Wikipedia corpus have achieved f-measure of 72.27%, 80.79% and 82.79% respectively. All gained obvious improvements over the unigram UMLS lookup results. Despite recall could slightly decrease, POS tags and Wikipedia corpus can make good contributions to

improve the precision. We keep only adjectives and nouns, thus verbs or adverbs like "drink" and "ago" which are wrongly recognized as medical entities can be removed. On the other hand, some short words are sometimes labeled with semantic types during UMLS lookup due to the ambiguous gene or protein names, such as "tg" and "dr". In these cases, the Wikipedia corpus is helpful to solve this problem. The vocabulary generated from Wikipedia corpus using the Word2vec tool can filter out non-standard oral expressions, like word abbreviations "b4" and "yrs". Some infrequent words, such as names "mich" or "lisa" can also be removed. NP chunking is proved to be the most effective for improving recall because it can recognize multi-word entities.

The results indicate that combining all the methods achieves the best effect, with the highest precision 86.56% and f-measure 82.79%. Although recall 80.20% is 2.21% lower than the result of NP chunking, it is still competitive and has a great improvement comparing to the unigram baseline.

4.4 Supervised Conditional Random Fields Results

Beside the unsupervised UMLS lookup method, we also wonder the effectiveness of supervised machine learning methods to further enhance the performance. Therefore, we utilized a conditional random fields model with various features (see Section 3.6.2) to perform the supervised sequence labeling task. In order to investigate the contribution power of the UMLS, NP chunking and clustering features respectively, we set the result of the combination of word, part-of-speech and context features as the CRF baseline and conduct leave-one-out feature experiments.

Additionally, we also performed different K values to compute word embedding k-means clustering and investigated their performance. In the experiments, we optimize the clustering effect on the tuning set, and evaluate it on the testing set. We run 4 times for each K value because the random selection of K centroids could cause convergent uncertainty. After this, the optimal one is chosen as the final result of each K value. Table 4.4 presents the results of k-means clustering. It shows the result achieves the best performance on the tuning set when K = 200, thus we use this K value for the testing set as well.

Table 4.4 and Table 4.5 are the CRF results with different features, among which the UMLS feature makes the most significant contribution to recall and the NP chunking feature has a great improvement on precision. The UMLS feature improves the recall by as high as 14% and as result the overall f-measure is improved by 6.11% up to

K values	Tuning set	Testing set
CRF NP chunking	84.19%	91.12%
K = 100	84.65%	91.69%
K = 200	**85.25%**	91.61%
K = 300	84.99%	91.59%
K = 400	85.19%	**91.71%**

TABLE 4.4: CRF F-measure with Different K Values

Configuration	CRF features	Precision	Recall	F-measure
UMLS baseline		85.56%	80.20%	82.79%
1	CRF baseline	96.92%	71.46%	82.27%
2	1 + UMLS	91.94%	85.09%	88.38%
2.1	1 + NP chunking	**97.73%**	76.07%	85.55%
2.2	1 + Clustering	96.89%	70.89%	81.87%
3	2 + NP chunking	93.65%	88.72%	91.12%
4	3 + Clustering	94.16%	**89.20%**	**91.61%**

TABLE 4.5: Supervised CRF Results (where K = 200)

88.38%. While applying the NP chunking feature alone makes recall decrease with 9%, it achieves the highest precision reaching to 97.73%.

For the clustering feature, it does not work well separately, but it can improve both precision and recall with the combination of UMLS and NP chunking features. As adding each feature, the overall performance is increasing as well. The combination of all features obtains the best f-measure of 91.61%. Furthermore, the results present that different K values have little effect on the final results, but all of them can increase the f-measure. The average contribution of clustering feature to f-measure is around 0.5%.

4.5 Discussion

Our experiment results prove that the performance of unigram UMLS lookup can have great improvements by applying POS tags and combining external n-gram noun phrase chunking. Lookup in a Wikipedia corpus can make additional improvements on the precision by removing non-standard use of language and infrequent words. The final f-measure of UMLS lookup is 82.79%, which provides a relatively high performance of medical entity recognition. From the results, it can be further improved by nearly 10% up to 91.61% by applying the conditional random fields model with word features, context features, the POS feature, and the features derived from UMLS lookup, NP chunking and word embedding clustering. The

UMLS feature contributes the most to the CRF model. While word embedding clustering has less effect on the result than other features, it also enhances the performance by around 0.5%. The different K values has little influence on the final result, but this somehow reflects the clustering result is stable

The CRF baseline result (82.27%) is comparable to the final performance of UMLS lookup (82.79%), while one advantage of the unsupervised UMLS lookup method over the supervised CRF method is that it does not need any labeled training data, which requires large labor work especially for the patient forum data. Another interesting fact is that the overall performance on the tuning set is only around 85%. The different results of tuning set and testing set suggest that the performance of CRF methods depends on the quality of training data. We believe this is mainly because the CRF method does not make a good prediction of unknown words for the training set. Patient forum data is changing over the time, thus the training dataset needs to be changing as well to maintain the same high standard performance. Nevertheless, our approach demonstrates that n-gram chunking does improve the performance of the UMLS lookup method and the CRF method with the right combination of features is needed to obtain the best performance of medical entity recognition from patient forum data.

4.6 Error Analysis

Although implementing CRF methods can achieve the best result, there are still problems left behind. Therefore, we investigate the possible causes of false positives and false negatives for the final results. According to our observations, we summarize three primary reasons for the errors.

- **Error caused by spelling mistakes** Spelling mistakes is one cause for the errors. Due to user-generated misspellings, some words like mistaken drug names can not be found in the UMLS database, thus they do not obtain UMLS features and have high possibility to be neglected by the approach. The error examples are “Sorafenibe” and “Nexvar”. However, some misspelling words can be recognized if they have same error samples in the training dataset.

- **Error caused by UMLS lookup** Most of the errors are caused by UMLS lookup. Some short or ambiguous words which can be found in the UMLS database are not effectively excluded by the system. One possible reason is that UMLS feature plays the most contributing feature for the CRF model, thus the system regards those words as right medical entities. On the other hand, around 75% of false negatives, including both words and noun phrases, if they are not in

the UMLS database or does not have predefined semantic types, are likely to be ignored by the system. UMLS lookup is an effective method for medical entity recognition, although with the help of noun phrase chunking and clustering, the problems could be alleviated, it also remains unresolved words and noun phrases. The error examples of false positives include "hunger", "dr" and "orange". For false negatives, error examples are "barefoot", "rego" and noun phrase "mitotic rate".

- **Error caused by unknown words** Some entities are in the testing set but not in the training set can not be recognized by our approach. This is mainly because the inherent limitations of machine learning methods. The model should "learn" in the first place and then make predictions for the testing set. Thus the performance of our CRF model depends on the quality of training set. If the training set is expanded, this cause could be solved. Error examples are "whipple" and noun phrase "hand foot syndrome".

According to the error analysis, we can make improvements by mainly two ways. First, we can improve the accuracy of UMLS lookup results. This can be done by applying word stemming or word disambiguation. Spelling correction could also be useful to correct mistaken words and improve the accuracy. Second, expand the size of training set could be helpful to reduce the number of unknown words, and thus improve the training performance.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this paper, we have proposed a novel approach for medical entity recognition from patient forum data and attained the overall F1 of 91.61%. Our approach includes both unsupervised UMLS lookup and supervised CRF methods, either of which has achieved relatively comparable results comparing to the existing MER methods.

With the unsupervised UMLS lookup method, we achieved an F1 of 82.79%. Our method proves that the performance of unigram UMLS lookup is not sufficient for medical entity recognition, the F1 of which is only 70.80%. Applying POS tags and n-gram noun phrase can have obvious improvements over the unigram results by filtering out wrongly recognized words and detecting multi-word terms. The UMLS is a powerful medical knowledge source for MER tasks, but it can not effectively solve the issues with word abbreviation and ambiguity, caused by non-standard use of language and infrequent words. We found that applying a vocabulary set of Wikipedia corpus which is generated by Word2vec tool can have help to avoid the problems and improve the accuracy.

Supervised conditional random fields method can have further improvements of the UMLS lookup results. As for this method, we explored the effectiveness of UMLS, NP chunking and clustering features separately, and found each feature can give improvements and contribute the final performance. The UMLS feature derived from the results of UMLS lookup is the most crucial feature in the CRF model which contributes a significant rise of the overall results. NP chunking can further have an impact on the precision based on the UMLS feature. Moreover, we also utilized unsupervised k-means clustering as a feature combined with the CRF model. The

results show that the performance of different k values is stable, while the effectiveness of the clustering feature does not have prominent improvements as UMLS and NP chunking features.

Our results also suggest that the performance of the supervised CRF model highly relies on the quality of the training set. If there are many different words in the training set and testing set, the performance might be very poor. Therefore, although the supervised method can achieve an almost 10% higher result over the UMLS lookup method, unsupervised UMLS lookup is more favorable for medical entity recognition if there are no suitable training datasets available. In order to reach a higher performance of MER, a supervised CRF model with right combination of features is indispensable.

Furthermore, the results of UMLS lookup are associated with the selection of UMLS semantic types. There is a trade-off of semantic types. Fewer semantic types can decrease the performance of UMLS lookup because it might decrease the recall, while superfluous semantic types might otherwise lower the precision. Therefore, the right selection of semantic types is of great importance. Also for this reason, we believe our approach is with flexibility and compatibility. It can adapt to other kinds of forum datasets as well because of the custom selection semantic types.

5.2 Future Work

Our method gives the insights of medical entity recognition from forum data. The result looks promising, and it still worth looking into the following improvements in the future.

First, we can expand the size of our data to further improve the model. In this paper, the size of training set and testing set are relatively small, thus there might remain some hidden problems. For example, the Wikipedia corpus used to generate k-means clustering might become less effective and insufficient to improve the performance. In addition, if the vocabulary size of Wikipedia corpus is small, it will filter out correct entities during the UMLS lookup. Moreover, increasing the amount of the training set can reduce the number of unknown words and improve the training quality.

Second, word normalization. We did not perform word normalization in this paper, while this can be useful to solve the issue with mistake spelling and enhance the overall performance.

Third, word sense disambiguation. According to the error analysis, we have residual problems that some words might have ambiguous meanings in different context, which can not be effectively recognized by our approach. Therefore, applying word sense disambiguation could be a possible way to solve the issue.

Furthermore, tuning experiment settings of the CRF model to increase compatibility. Feature extraction is of great importance to a CRF model. Due to the size of the datasets used, we used limited features. Even though the proposed features in this paper are sufficient to train the model, but as the increasing of the data size, the choices of features could be various. For example, we used context feature with one following and one previous words in this paper, while for a larger dataset, the context feature is becoming more important, and the model might need more context information. Therefore, the experiment settings of our approach might need to be modified to become more general and compatible.

Bibliography

[1] Asma Ben Abacha and Pierre Zweigenbaum. "A hybrid approach for the extraction of semantic relations from MEDLINE abstracts". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer. 2011, pp. 139–150.

[2] Asma Ben Abacha and Pierre Zweigenbaum. "Automatic extraction of semantic relations between medical entities: a rule based approach". In: *Journal of biomedical semantics* 2.5 (2011), S4.

[3] Asma Ben Abacha and Pierre Zweigenbaum. "Medical entity recognition: A comparison of semantic and statistical methods". In: *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics. 2011, pp. 56–64.

[4] Kumar Aman et al. "Understanding Medical Named Entity Extraction in Clinical Notes". In: *International Conference on Health Informatics and Medical Systems*. BCL Technologies. 2014, pp. 201–204.

[5] Alan R Aronson. "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.

[6] Alan R Aronson and François-Michel Lang. "An overview of MetaMap: historical perspective and recent advances". In: *Journal of the American Medical Informatics Association* 17.3 (2010), pp. 229–236.

[7] Andreea Bodnari et al. "A Supervised Named-Entity Extraction System for Medical Text." In: *CLEF (Working Notes)*. 2013.

[8] Aaron M Cohen. "Unsupervised gene/protein named entity normalization using automatically extracted dictionaries". In: *Proceedings of the acl-ismb workshop on linking biological literature, ontologies and databases: Mining biological semantics*. Association for Computational Linguistics. 2005, pp. 17–24.

[9] Kerstin Denecke and Wolfgang Nejdl. "How valuable is medical social media data? Content analysis of the medical web". In: *Information Sciences* 179.12 (2009), pp. 1870–1880.

[10] Vijay Garla et al. "The Yale cTAKES extensions for document classification: architecture and application". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 614–620.

[11] Ying He and Mehmet Kayaalp. "Biological entity recognition with conditional random fields". In: *AMIA Annual Symposium Proceedings*. Vol. 2008. American Medical Informatics Association. 2008, p. 293.

[12] Geoffrey E Hinton. "Learning distributed representations of concepts". In: *Proceedings of the eighth annual conference of the cognitive science society*. Vol. 1. Amherst, MA. 1986, p. 12.

[13] Ferdaous Jenhani, Mohamed Salah Gouider, and Lamjed Ben Said. "A hybrid approach for drug abuse events extraction from Twitter". In: *Procedia computer science* 96 (2016), pp. 1032–1040.

[14] Ning Kang et al. "Knowledge-based extraction of adverse drug events from biomedical text". In: *BMC bioinformatics* 15.1 (2014), p. 64.

[15] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[16] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[17] Xiao Liu and Hsinchun Chen. "Identifying adverse drug events from patient social media: A case study for diabetes". In: *IEEE Intelligent Systems* 30.3 (2015), pp. 44–51.

[18] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[19] Bethesda (MD): National Library of Medicine (US). *UMLS® Reference Manual [Internet]*. 2009. URL: https://www.ncbi.nlm.nih.gov/books/NBK9676/.

[20] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[21] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[22] Azadeh Nikfarjam et al. "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features". In: *Journal of the American Medical Informatics Association* 22.3 (2015), pp. 671–681.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation." In: *EMNLP*. Vol. 14. 2014, pp. 1532–1543.

[24] Guergana K Savova et al. "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications". In: *Journal of the American Medical Informatics Association* 17.5 (2010), pp. 507–513.

[25] Christian Siefkes. "A comparison of tagging strategies for statistical information extraction". In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics. 2006, pp. 149–152.

[26] Maria Skeppstedt. "Enhancing Medical Named Entity Recognition with Features Derived from Unsupervised Methods." In: *EACL*. 2014, pp. 21–30.

[27] Parikshit Sondhi et al. "Shallow information extraction from medical forum data". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics. 2010, pp. 1158–1166.

[28] Vıctor Suárez-Paniagua, Isabel Segura-Bedmar, and P Martínez. "Word embedding clustering for disease named entity recognition". In: *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*. 2015, pp. 299–304.

[29] Buzhou Tang et al. "Evaluating word representation features in biomedical named entity recognition tasks". In: *BioMed research international* 2014 (2014).

[30] Hongkui Tu et al. "When MetaMap Meets Social Media in Healthcare: Are the Word Labels Correct?" In: *Information Retrieval Technology*. Springer, 2016, pp. 356–362.

[31] Hanna Wallach. "Efficient training of conditional random fields". PhD thesis. Master's thesis, University of Edinburgh, 2002.

[32] Hanna M Wallach. "Conditional random fields: An introduction". In: *Technical Reports (CIS)* (2004), p. 22.

[33] Shaodian Zhang and Noémie Elhadad. "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts". In: *Journal of biomedical informatics* 46.6 (2013), pp. 1088–1098.

[34] Yaoyun Zhang et al. "Chemical named entity recognition in patents by domain knowledge and unsupervised feature learning". In: *Database* 2016 (2016).