Health Information Extraction from Social Media

by

Azadeh Nikfarjam

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2016 by the
Graduate Supervisory Committee:

Graciela Gonzalez, Chair
Robert Greenes
Matthew Scotch

ARIZONA STATE UNIVERSITY

August 2016

ABSTRACT

Social media is becoming increasingly popular as a platform for sharing personal health-related information. This information can be utilized for public health monitoring tasks such as pharmacovigilance via the use of Natural Language Processing (NLP) techniques. One of the critical steps in information extraction pipelines is Named Entity Recognition (NER), where the mentions of entities such as diseases are located in text and their entity type are identified. However, the language in social media is highly informal, and user-expressed health-related concepts are often non-technical, descriptive, and challenging to extract. There has been limited progress in addressing these challenges, and advanced machine learning-based NLP techniques have been underutilized. This work explores the effectiveness of different machine learning techniques, and particularly deep learning, to address the challenges associated with extraction of health-related concepts from social media. Deep learning has recently attracted a lot of attention in machine learning research and has shown remarkable success in several applications particularly imaging and speech recognition. However, thus far, deep learning techniques are relatively unexplored for biomedical text mining and, in particular, this is the first attempt in applying deep learning for health information extraction from social media.

This work presents ADRMine that uses a Conditional Random Field (CRF) sequence tagger for extraction of complex health-related concepts. It utilizes a large volume of unlabeled user posts for automatic learning of embedding cluster features, a novel application of deep learning in modeling the similarity between the tokens. ADRMine significantly improved the medical NER performance compared to the baseline systems.

This work also presents DeepHealthMiner, a deep learning pipeline for health-related concept extraction. Most of the machine learning methods require sophisticated task-specific manual feature design which is a challenging step in processing the informal and noisy content of social media. DeepHealthMiner automatically learns classification features using neural networks and utilizing a large volume of unlabeled user posts. Using a relatively small labeled training set, DeepHealthMiner could accurately identify most of the concepts, including the consumer expressions that were not observed in the training data or in the standard medical lexicons outperforming the state-of-the-art baseline techniques.

DEDICATION

To Ehsan, my love and companion all the way, to my parents for their constant support and encouragement, and to my lovely Arman.

ACKNOWLEDGMENTS

I owe my gratitude to all those people who have made this dissertation possible. First, I am sincerely grateful to all members of my committee – Dr. Graciela Gonzalez, Professor Robert Greenes and Dr. Matthew Scotch. I am especially thankful to Professor Greenes for his significant guidance in defining my research scope and for his helpful advices during the course of my graduate studies at ASU. I am sincerely grateful to Dr. Scotch for supervising this dissertation and for his insightful comments and advices that have been a significant help for completing this work. My deepest gratitude is to my advisor, Dr. Graciela Gonzalez for her constant support and valuable guidance all these years. She helped me to choose the right path for my research, and at the same time trusted my effort and gave me the freedom to explore on my own. I truly thank her for her encouragements, her patience and for the opportunities provided me to grow.

I would like express the most sincere appreciation to Professor William Johnson, my graduate program advisor, for his constant encouragements and valuable advices.

I am particularly grateful to Dr. Nigam Shah for providing me the opportunity to collaborate with his lab, for providing the resources needed for my research, and for the valuable feedback of him and his lab members on my work.

I am especially thankful to Dr. Robert Leaman, Dr. Ehsan Emadzadeh and Dr. Abeed Sarker for the constructive suggestions and valuable feedbacks on my research. I owe thank to my collaborators and lab members, Karen O'Connor, Pranoti Pimpalkhute, Rachel Ginn, Apurv Patki, Swetha Jayaraman, Tejaswi Upadhyaya, Dr. Karen Smith, Dr. Michael Paul, Dr. Ioannis Korkontzelos and Professor Sophia Ananiadou.

I acknowledge the support of NIH National Library of Medicine grant number NIH NLM 5R01LM011176.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF EQUATIONS

Equation Page

1 INTRODUCTION

The widespread use of social media has provided a platform for internet users to share experiences and opinions, and has turned social networking websites into valuable sources of information. The massive amount of user-generated content can be used for various tasks such as measuring political sentiments [1], predicting stock market trends [2] or general sentiment analysis [3]. Similarly, social media can be used for tracking public health trends, since people tend to share information about events and details in their life such as their health status. Although knowing about a few individuals' health may not seem interesting, millions of health-related messages can reveal important public health issues. For instance, the user posts can be used for tracking the spread of contagious diseases such as influenza [4–6], monitoring the time and geographical locations of diseases [7], studying the treatment outcomes [8], finding correlations between symptoms and treatment choices [9], and discovering the potential adverse or beneficial effects of medications [10,11]. Natural language processing techniques can be used to extract useful information from social media.

Problem Statement

One of the important and fundamental tasks in most language processing pipelines is the identification and extraction of relevant concepts (also referred to as named entity recognition (NER)). This dissertation focuses on health-related concepts, mentioned in social media postings, which is a relatively recent problem in social media analysis [9,10]. Some of the general challenges in extraction of health-related concepts from social media are listed as follows:

- Consumers do not always use technical terms for reporting health-related information and instead use alternative creative terms, explanations of the symptoms and idiomatic expressions. For example, consider “feel like I was in a fog” or “half awake-half asleep state” which are used in user reviews about drugs in reporting adverse effects. We refer to these creative and descriptive user expressions as “consumer expressions”.
- The available dictionaries do not include all the possible variations of a concept (especially consumer expressions).
- The consumer sentences are usually informal which do not follow grammatical rules, and include spelling and structural problems that could potentially cause poor performance of existing language processing tools such as part-of-speech taggers and parsers.
- The concept extraction solutions which are mainly based on observed keywords in the sentences (specifically the lexicon-based methods or machine learning classifiers which use surface lexical features) are not successful in extracting complex concepts. For instance, a matched entry from the ADR lexicon in a user review is not necessarily an adverse effect, as it can be instead a mention of an indication (reason to use the drug) or a beneficial effect. Similarly, in disease surveillance studies based on social media, a *flu* mention in a tweet is not always reporting the disease infection, and can be about the concerns of the user about the flu.

Thus, extracting complex concepts (consumer expressions) from user-generated sentences is more difficult compared to named entity recognition in other corpora such as news or biomedical literature.

Lexicon-based methods that primarily rely on string comparison techniques, usually perform well in targeting the names of people, geographical locations, genes and proteins, whereas, in social media health-related concepts are descriptive and complex to extract. To address some of the limitations of the lexicon-based methods, machine learning methods are generally applied [11–13]. We proposed and published a pattern-mining approach [11] that automatically learns concept extraction patterns from the training data. We then used that patterns to extract mentions of adverse drug effects from user posts in health related websites. The method could locate many of the challenging consumer expressions in the user reviews. However, there are some limitations associated with the pattern-based method that prevent it from being a stand-alone solution for this task. One of the main challenges is its dependence on the size of the training data, since it identifies an extraction pattern only if enough matching sentences are observed in the training data. This makes it difficult to locate concepts expressed in less frequent and more complex sentences such as user posts on Twitter. There has been limited progress in addressing these challenges, and thus far, advanced machine learning-based NLP techniques have been underutilized.

Considering these limitations, using a supervised machine learning-based method such as Conditional Random Fields (CRF) [14], seems to be an effective solution for this task. In fact, CRF is the state-of-the-art method used for concept extraction from both formal [15,16] and informal text [13,17]. However, supervised methods are still dependent on

large volumes of labeled training data, and this dependency is aggravated when dealing with social media content due to its noisy and informal nature. Moreover, informal text affects one of the building blocks of machine learning methods which is feature engineering. Most of the conventional machine learning methods require sophisticated task-specific manual feature design. Some of the features, such as capitalized words, are very reliable in NER methods for corpora such as news articles, which tend to follow orthographic and grammar rules more closely (are well-formatted), but are unreliable for social media. In addition, many of the classification features are usually calculated based on the output of other language processing tools such as part-of-speech taggers, shallow and deep grammar parsers which are trained on well-formatted corpora and their performance is usually weak for the noisy and informal text.

Recently deep learning techniques, a new class of machine learning methods based on non-linear information processing, have shown remarkable success in automatic feature engineering and have revolutionized the methods for computer vision [18] and speech recognition [19]. Deep learning methods have also achieved near state of the art results in NLP tasks, including chunking and Named Entity Recognition (NER) in well-formatted domains such as news or Wikipedia content [15,20]. However, thus far, deep learning techniques are relatively unexplored for biomedical text mining and, in particular, medical concept extraction from social media.

The primary aim of this dissertation is to propose a novel natural language processing solution that address most of the abovementioned challenges in extraction of medical concepts from user posts in social media. As a case study, we focus on adverse drug reaction (ADR) mentions, however, the proposed techniques are general and can easily

be adopted for other concepts such as mentions of diseases, treatments or health outcomes, as well as concepts in other domains.

Hypothesis and Contributions

Our first aim is to evaluate the effectiveness of incorporating unsupervised learned features on the performance of a supervised machine learning system for extraction of health-related concepts from social media. We hypothesized that adding unsupervised learned features to an existing supervised system improves the concept extraction performance. We present ADRMine, a machine learning sequence tagger for concept extraction from social media. We explore the effectiveness of various contextual, lexicon-based, sentiment, grammar and semantic features. We propose "embedding cluster features", a novel application of deep learning in modeling the similarity between the tokens in the NER systems. These features are based on word clusters generated from pre-trained word representation vectors (also referred to as word embeddings [15]), that are learned from more than 3.5 million unlabeled user sentences, using a deep learning technique.

Our second aim is to propose a deep learning system to evaluate the effectiveness of automatic feature learning for health-related concept extraction from social media. We introduce DeepHealthMiner, a deep learning pipeline that uses a feedforward neural network for the extraction task. The neural network classifier automatically learns classification features and does not require manual feature engineering. Considering the noisy and informal nature of the user posts that intensifies the challenge of the manual feature design, we hypothesized that DeepHealthMiner, with automatic feature learning

would address many of the abovementioned challenges associated with social media data, and would accurately identify most of the ADR mentions, including the consumer expressions that are not observed in the training data or in the standard ADR lexicons. We expect that the proposed methods improve the performance of medical concept extraction compared to state-of-the-art CRF classifiers.

Furthermore, we hypothesized that ADRMine and DeepHealthMiner that both apply deep learning techniques and utilize the unsupervised learned word embeddings would diminish the need for large amounts of labeled data, which are generally required to train supervised machine learning classifiers.

This dissertation is organized as follows. Chapter 2 provides the background and fundamentals, and includes the related literature. Information about the data collection and annotation is provided in Chapter 3 and is followed by a detailed explanation of the proposed methods in Chapter 4. The results are reported and discussed in Chapter 5 followed by the conclusions in Chapter 6.

2 BACKGROUND

Social Media and Public Health

The increasing popularity of internet-based media consumption has changed many aspects of the human life including communications and the way that people share or seek information. People tend to use the social media to share information about the events or the details of their personal life including their health. In a survey in 2012, Fox and Duggan [21] demonstrated that 72% of internet users have looked online for health information in the past year including searches related to serious conditions, general information, and minor health problems. The valuable health-related content accessible from tweets [9], online search query logs [22] or the user posts in the forums can be used for several purposes including measurement of patients' satisfaction of services [24], infectious disease surveillance [4–6], health outcome measurement [8] and drug adverse reaction detection [10,11].

Adverse Drug Reactions and Pharmacovigilance

An adverse drug reaction is defined as unwanted or harmful reaction experienced after using a drug under normal conditions of use and suspected to be related to the drug [23]. ADRs are a major public health concern and are among the top causes of morbidity and mortality [24]. According to a systematic review of twenty-five prospective observational studies including 106,586 patients who were hospitalized, approximately 5.3% of all hospital admissions are associated with adverse drug reactions, with higher rates (a median of 10.7%) reported for elderly patients [25]. If ADRs were ranked as a disease by cause of death, it would be the fifth leading cause of death in the United States [26]—

ahead of pulmonary disease, diabetes, AIDS, pneumonia, accidents, and automobile deaths. The economic impact of ADRs is also important: approximately $136 billion is spent annually on treating ADRs in the U.S., with other nations facing similar difficulties [27,28].

Some of the adverse effects are discovered during Phase III trials, however, there are some that are only revealed after a long time use or at the end of treatments. New ADRs also appear when the drug is used by groups of patients not included in the trial (for example, children, pregnant women, elderly or patients with chronic diseases).

Post-market drug safety surveillance is therefore required to identify potential adverse reactions in the larger population to minimize unnecessary, and sometimes fatal, harm to patients.

Post Market Drug Safety Surveillance

Most of the drug safety monitoring activities are based on reports by clinicians. However, there are studies that demonstrate the potential contribution of consumers' reports in discovering adverse effects [10,29–31]. For instance, Egberts et al. [30] performed a retrospective study to reveal the value of consumer reports. The authors assessed data gathered from a Dutch phone call service that allowed patients to consult with a pharmacist regarding the side effects of drugs. The time at which the first report of previously unrecognized adverse reactions was received from a patient was compared with the time of receipt from the first health-professional report to the regulatory authority database of the same reaction. On average, the first professional report was received nine months later than the report by a patient [30].

Spontaneous reporting systems (SRS) are surveillance mechanisms supported by regulatory agencies such as the Food and Drug Administration (FDA) in the U.S., which enable providers and patients to directly submit reports of suspected ADRs. When compared to reports from other providers, patients' reports have been found to contain different drug-ADR pairs, contain more detailed and temporal information, increase statistical signals used to detect ADRs, and increase the discovery of previously unknown ADRs [32–35]. However, under-reporting limits the effectiveness of SRS. It is estimated that over 90% of ADRs are under-reported [36].

To augment the current systems, there are new ways to conduct pharmacovigilance using expanded data sources — including data available on social media sites, such as Twitter [37,38], or health-related social networks, such as DailyStrength [39]. While a few individuals' experiences may not be clinically useful, thousands of drug-related posts can potentially reveal serious and unknown ADRs. Figure 1 shows examples of ADR-relevant user postings from Twitter (a) and DailyStrength (b), with labeled mentions.

a) *#**Schizophrenia**indication #Seroquel did not suit me at all. Had severe **tremors**ADR and **weight gain**ADR.*

b) *I felt awful, it made my **stomach hurt**ADR with bad **heartburn**ADR too, **horrid taste in my mouth**ADR tho it does tend to clear up the **infection**Indication.*

Figure 1: Examples of user posts about drugs in Twitter (a) and DailyStrength (b).

Pharmacovigilance from Social Media

Much research is dedicated to text mining approaches for identification of ADRs from clinical documents such as electronic health records [40–42] and medical case reports [43,44]. Harpaz et al. [45] provided a through survey on the techniques for

pharmacovigilance, utilizing various resources such as electronic records, spontaneous adverse drug reporting systems and biomedical literature. There are relatively fewer studies that investigate the language processing techniques for extraction of drug effect mentions from user comments in social media. In our previous study [46], we performed a methodical review to characterize the different approaches to ADR detection/extraction from social media, and their applicability to pharmacovigilance.

Leaman et al. [10] and Nikfarjam et al. [11] were the first to propose text mining methods for pharmacovigilance. Using a lexicon-based approach, Leaman et al. [10] studied the potential value of social media postings and demonstrated that user comments contain extractable information relevant to pharmacovigilance. Alternatively, we proposed a pattern-based approach to address some of the limitations of the lexicon-based method by capturing the underlying syntactic and semantic patterns from user reviews [11].

Chee et al. [47] classified user posts on health forums to predict the candidate FDA (U.S. Food and Drug Administration) watchlist drugs for further investigation with regards to drug safety. The authors classified the drugs as watchlist using a machine learning method which classified the drugs based on the whole content about the drug; however, they did not extract the ADR mentions from the user posts**.** A study by Wicks et al. [48] in the research center of PatientsLikeMe website suggested that data provided by patient communities over the internet can be useful in speeding up the clinical discoveries about the drugs for rare diseases.

Most of the previous text mining solutions [49–52] for adverse effect extraction from social media are based on dictionary lookup, whereby words in sentences are compared to a lexicon of adverse effects. Yates and Goharian [50] used a custom ADR dictionary

and limited lexical patterns for analysing the patient comments on a list of popular cancer drugs from three different health websites: askapatient.com, drugs.com and drugratingz.com. Authors evaluated the extracted ADRs for a drug with SIDER database [53] information, to see if the extracted ADR was known or unknown.

There are studies [50,52,54] that have explored patient discussions for extraction of useful drug safety information. Yang et al. [55] used the user discussions on MedHelp[1] website and utilized a dictionary lookup approach to locate ADRs from a lexicon. Sampathkumar et al. [54] proposed a similar lexical approach for recognizing the drugs and the ADR mentions from user messages on Medications.com. These lexicon-based methods [54], [55] simply considered any matched phrase as an adverse effect and did not propose any solutions for distinguishing the matched lexicon entries that are not ADR concepts (and instead could be indications). The studies that work on patient discussions require to use a relationship extraction technique to map the ADRs to the related drugs. However, in this dissertation, we only focus on concept extraction techniques since we are dealing with short messages such as user tweets. Also, DailyStrength website provides a page for each drug, and all the related comments for the drug are added in that page. Similarly, we target the tweets based on keyword search and only analyze the tweets that include the drug names. Therefore, we do not need to extract the drug names for this task. However, our proposed system can be used as an independent concept extraction module in other processing pipelines including relationship extraction systems.

[1] http://www.medhelp.org/

Name Entity Recognition Techniques

The methods for named entity recognition can be divided into three main categories [56]: dictionary based, rule based, and machine learning techniques.

The dictionary approaches, typically use a large list of entity names including instances of different entity types and locate the entities of interest in natural language text. The main advantage of dictionary methods is that the extracted mentions are also normalized to their corresponding concept ID in the dictionary. However, the entity names in the dictionaries are limited and they can never include all possible variations of one entity, particularly when considering the creative and complex mentions of medical concepts found in user posts. In addition, these techniques perform poorly in identifying the mention entity types for the cases where one dictionary entry belongs to more than one entity type (e.g. ADR and Indication). Despite these limitations, the dictionary methods are very commonly used in biomedical domain, particularly for information extraction tasks from social media.

The rule based methods apply a set of patterns, often using regular expressions, to extract the entities. The primary advantage of a rule based technique is that it does not require a long list of entity names. The second advantage of this method is that the system decisions for extracting a concept can easily be justified; the rule that was triggered can be shown to users, explaining why the entity was extracted [57]. On the other hand, it has the disadvantage of not generalizing well. Although the rules usually result in high precisions, they can only cover limited instances. In addition, tuning the parameters and manually modifying the rules to achieve higher performance is very time consuming and usually result in very specific rules with low recall.

Machine learning based methods have long been used in NER systems for general domain. They have also successfully been applied in biomedical NER systems [16]. The machine learning methods for NER fall into three main categories: supervised, semi-supervised and unsupervised learning methods [58]. Supervised machine learning methods are commonly used for this task and typically require a large annotated data set to build a classification model. The system learns to extract entities using a set of positive and negative examples. The examples are represented to the system using a set of features. Early methods used instance classifiers such as Naive Bayes or support vector machine to classify the tokens [57]. Later systems used sequence tagging classifiers including hidden Markov models (HMMs), and maximum entropy Markov models (MEMMs). The success of the most machine learning methods depends on the representation of the features. Machine learning developers usually define a set of hand crafted features for every specific task. The system learns to extract based on these features and usually generalize much better compared to rule based methods. However the main drawback of supervised learning methods is the dependency to large number of annotated instances, while annotation in general is a labor-intensive and costly process. In addition, the process of feature engineering is usually challenging particularly for biomedical extraction task that require some levels of domain knowledge. Also every task usually requires specific feature engineering. Deep learning is a new class of machine learning methods that aims at automatic feature learning methods and address some of the above mentioned challenges [59]. Deep learning techniques are very recent and are relatively unexplored in the biomedical domain. More details about deep learning methods is presented in Deep Learning Overview section.

Concept Extraction From Social Media

The type of text in our concept extraction task is very similar to user posts in other social media such as Twitter. With increasing popularity of micro-blogging, the research interest in automatic analysis and entity (concept) extraction from tweets and similar content has increased. As mentioned earlier, this type of text is informal and noisy in nature, therefore, the traditional entity recognition methods that are developed for analyzing documents such as news perform poorly for social media content [13,60,61]. To address part of the challenges in social media, unsupervised concept extraction solutions using web-based resources such as Wikipedia is becoming very popular. Michelson and Macskassy [60] proposed a novel approach for discovering users' topics of interest from Tweets. The method requires named entity recognition in the first step. The authors simply took any capitalized, non-stop word as a possible entity and used a novel disambiguation approach to find a mapping for the entity in Wikipedia. Similarly, Li et al. [61] used an unsupervised approach for NER utilizing Wikipedia and the web N-gram corpus [62] to find the candidate named entities and then disambiguated the candidates based on a scoring function which considers the context in the tweet, Wikipedia and the web N-gram corpus for calculating the score. Furthermore, there are supervised entity extraction approaches that are tailored for social media characteristics [13,63]. Ritter et al. [13] demonstrated that existing language processing tools for POS tagging, chunking and NER do not perform well when applied to tweets. The authors proposed a new NLP pipeline, including tools specifically trained for Twitter. The NER module identifies companies, products, movies and other entity types, using conditional random fields and utilizes several engineered features such as contextual, orthographic

and dictionaries, plus the outputs of the Twitter-specific POS tagger and the chunker. The authors showed that the proposed NER system outperformed the existing systems around 50% in F-measure [13]. However, training the system required manual annotation of POS tags in 800 tweets; furthermore, the success of the proposed method is largely dependent to the available dictionaries and Wikipedia page titles. Given the earlier mentioned characteristics of the complex entities in our task, even if we assume that the Twitter-specific POS tagger and the chunker perform well on the drug review corpus, the dependency of the Ritter's method to the existing Wikipedia page titles and other online dictionaries, make it impractical for our concept extraction task.

Deep Learning Overview

Artificial Neural Networks

Artificial Neural networks (NN) are parallel computing systems that were designed for solving complex problems such as pattern recognition and optimization by mimicking the neural networks of the human brain. A biological neuron, as shown in Figure 2, is composed of a cell body and two tree-like branches, axon and dendrites. The neuron receives signals from other neurons through dendrites and transmits the generated signal to another set of neurons through the terminal branches. Inspired by the natural neural networks, the artificial neural network was introduced by McCulloch and Pitts [64]. As Figure 2 illustrates, it has a set of inputs, a processing unit, a binary threshold unit and a binary output. The weighted sum of the inputs is calculated, and then passed into a threshold unit such as a sigmoid function to generate the binary output (0 or 1) based on the sum value. The formal notation is shown in Equation 1, where x_i is the i^{th} input, w_i is

associated weight and *h* is an activation function such as sigmoid function that turns 1 if the input is larger than a threshold (e.g. >0.5) and 0 otherwise.

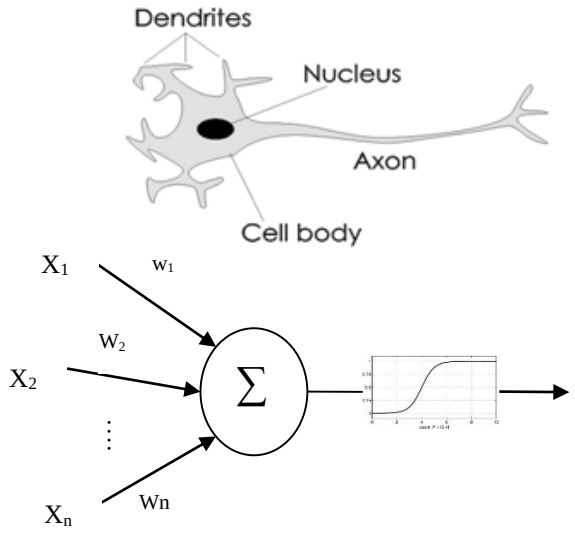


Figure 2: Schematic comparison between a biological neuron and an artificial neuron.[2]

$$y = h\left(\sum_{i=0}^{n} w_i x_i\right)$$

Equation 1: The activation function of a neuron.

A neural network is a layered network of these individual neurons, as shown in Figure 3, and usually consists of an input layer, an output layer and one or more hidden layers which transform the inputs into a more complex and abstract representation. The neurons of every layer are connected through weighted edges to the neurons of the next layer. These weights are the parameters of the network which are adjusted during the learning process.

[2] http://www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html

One of the main structures of neural networks is *feedforward* neural networks. In the feedforward networks the layers are connected in one direction without any loop in the network.

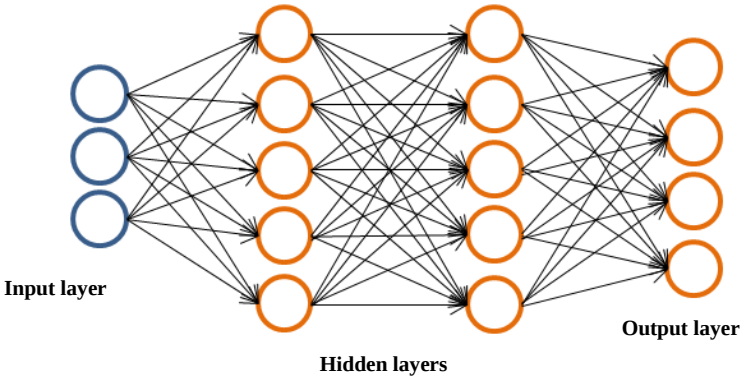


Figure 3: Example of a deep neural network.

Learning Process in Neural Networks

One of the characteristics of the intelligent systems is the ability to learn. Learning in NNs can be viewed as updating the connection weights between the nodes so that the network can optimally perform a specific task. One of the most common algorithms for NN training is the stochastic gradient descent which is a learning algorithm based on error derivatives. For learning the parameters of the model, a cost function is defined, which typically is the sum of the squared errors for all the training examples. The error for each training example is the difference between the desired value (d), and the generated output (y) by the network ($d - y$). During the learning process, the parameters of the model, the weights of the NN, get updated using gradient descent that works by calculating the partial derivatives of the cost function. The errors are back-propagated, and the individual weights are updated accordingly. More details about the neural networks and the learning process can be found in [65–67].

Deep Learning for Natural Language Processing

One of the main barriers in taking full advantage of processing capacities of computers in language processing is the difficulty of encoding the meaning of human language. When using machine learning methods in different classification tasks, one usually defines many manually engineered features which are speculated to be useful for representing the meaning (semantic) of words, phrases, sentences or documents. Representation learning is a set of methods that automatically discover features from raw data. Deep learning methods are representation learning methods with multiple levels of representation [59]. Deep learning methods typically use neural networks to transform the input with each hidden layers of the network, creating a more abstract representation of the input [68]. The idea of using neural networks for sequence prediction dates back to 1990 [69]. However, the first serious attempt in using NNs for the task of language modeling was proposed by Bengio [70] in 2001. Several other neural network-based models then were proposed [71,72], but much research was on theories and did not work on the practical problems when using those complicated models. At that time, the computational complexity of the neural networks was too high for the real problems. Bengio reported that training a neural net language model in 2001, took a week using 40 CPUs for a single training epoch, while 10 to 20 epochs were needed to be completed [73]. Since then, most of the research have been focusing on reducing the complexities of such models for practical purposes [74,75].

Word Embeddings

Most of the machine learning systems used for language processing tasks, regard words as atomic symbols represented with a vector. The size of the vector is very large (for example 500K or even 13M), usually equal to the size of the vocabulary in the corpus, while only one-dimension value in that space is "1" (the index of the word in the dictionary) and the rest of the values are "0". This way of representing the words is called "one-hot" or "one-of-V" representation (e.g. [0 0 1 0 … 0]). One of the major problems with one-hot representation is that it doesn't represent the similarity between the words. Therefore, if a word in the test sentence is not seen in the training set, although the similar words are present in the training data, the machine does not consider this similarity. Therefore, providing a dense representation that considers the similarity between the words has been an ongoing research in machine learning, and several solutions have been proposed such as latent semantic analysis (LSA) [76], Latent Dirichlet Allocation (LDA) [77], and recently neural network language models [78,79]. In all of the solutions, the goal is to find a dense and meaningful representation (e.g. [0.79, −0.17… 0.10, 0.34]). The dense representation vector of a word or phrase is also referred to as word embeddings [15].

Word embeddings are vector representations, with configurable dimensionality (usually 150 to 500), of words that are obtained as a side result of training an auxiliary task that is building a language model [80,81]. For training the language model, every word (w) in a sentence is considered as a training instance. Two common model architectures for learning distributed representations of words are CBOW (Continuous Bag-of-Words) and Skip-gram [75]. Both architectures use neural networks for training a language model. In

the CBOW architecture, the neural network learns to predict a word given its context in the sentence, while in the Skip-gram architecture (Figure 4), the neural network learns to predict the context of the given word. The context is typically defined as a few preceding and following words in a sentence. The input word is usually represented with one-hot encoding. The network learns the input word's embeddings by observing the context window of the word in several different sentences. Every context word is also represented using a one-hot vector. The network estimates the probability distribution of a number of previous and next words, located in a window of configurable size (equals to 8, in this study) around w (Figure 4). Word2vec [75] is a neural network classifier that can be trained to compute the word embeddings by learning the auxiliary task of language modeling. It first constructs a vocabulary from the input corpus. Training the language model is completely unsupervised and does not require any labeled examples.

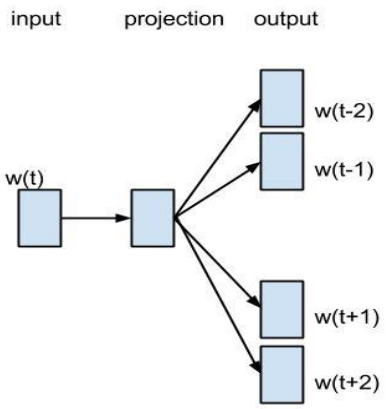


Figure 4: Neural network language model for learning word embeddings [75].

Using this architecture, the similar words that share the same neighbor word probability distribution (in the output layer) will be projected into similar vectors by the projection (hidden) layer of the network. In fact, the point in using the hidden layer is to do this

transformation and represent the similar words, with close vectors in the embedding space. It has been shown that both syntactic and semantic similarities can be modeled by using this approach in training the embeddings [82]. The similarities between the generated embedding vectors can be measured using vector similarity metrics such as cosine similarity score. More information about generating the embeddings can be found in the related papers [15,70,75].

Deep Learning techniques for NER

Here we briefly explain the neural network computations used in a typical classification problem. A named entity recognition problem can also be formulated as a classification problems, where every token is considered a classification candidate. The architecture of a simplified feedforward neural network is demonstrated in Figure 5. The simplified network architecture and notation is inspired by the approach used in [78,83].

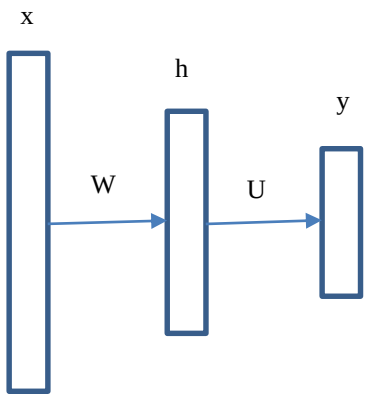


Figure 5: A simple feedforward neural network with one hidden layer.

The network has an input, a hidden and an output layer. The nodes in the input layer are fully connected to the hidden layer through the weighted edges (*W*). Similarly the nodes

in the hidden layer are connected to the output nodes with weighted edges (*U)*. A training candidate is a pair denoted as $(x,\ y)$, where x is a vector representing the input and y represents the class of x.

The hidden layer activation is defined as follows:

$$h = \sigma(Wx + b_1)$$

Equation 2: the activation function of the hidden layer [83].

Where:

- W $\in \mathbb{R}^{m \times n}$ is the weight matrix that connects the nodes in the input layer ($x \in R^m$) to the hidden layer,
- $b_1 \in \mathbb{R}^m$ is a vector of biases in the input layer,
- and $h \in \mathbb{R}^n$ is a vector of activation unit outputs corresponding to the nodes in the hidden layer. Typically, the activation function can be sigmoid or a hyperbolic tangent (tanh) or rectified linear unit (ReLU) [59]. If the input of an activation function is larger than a threshold the output value would be 1 and 0 otherwise.

The output layer is a softmax function defined as follows:

$$\hat{y} = softmax\ (hU + b_2)$$

Equation 3: The predictions in the neural network output layer [83].

- $U \in \mathbb{R}^{n \times K}$ is the weight matrix that connects the hidden layer to the output softmax layer; *n* is the number of hidden nodes and *k* is the number of nodes in the output layer,
- and $b_2 \in \mathbb{R}^n$ is a vector of biases in the hidden layer,
- The softmax function normalized the vector of output layer score to be in the range of (0,1) and add up to 1.

During training, for every training candidate the cross entropy loss is calculated. The goal of the neural network during is to minimize the cross entropy loss (Equation 4).

$$CE(y, \hat{y}) = -\sum_{i=1}^{k} y_i \log \hat{y}_i$$

Equation 4: the cross entropy loss function.

where:

- k is the number of possible classes (number of output nodes),
- y is the expected outcome,
- $\hat{y}$ is the predicted outcome.

The learning process is typically based on stochastic gradient dissent which was briefly explained in the previous section and more details can be found in [20,59].

Collobert and Weston [20] demonstrated the effectiveness of using deep neural networks for automatic feature learning for a number of natural language processing tasks such as POS tagging, chunking, and NER on well-formatted text such as news corpora. They trained a feedforward neural network for identifying named entities such as names of people, companies, and locations. They used the word embeddings learned from Wikipedia (Collobert-Weston embeddings), for representing the words. The authors demonstrated that, without adding any engineered features, a simple neural network NER system, similar to the abovementioned network, can reach the performance near the state-of-the-art. However, the effectiveness of deep learning is not studied for extraction of complex medical concepts from the informal and noisy content in the social media.

In summary, this chapter provided the background information about social media and health with a focus on pharmacovigilance from social media. We surveyed the named

entity recognition techniques and summarized the related research on health information extraction from social media. Finally the fundamentals about deep learning for NLP was presented.

3 DATA COLLECTION AND ANNOTATION

We collected user posts about drugs from two different social media resources: DailyStrength (DS) and Twitter. In this study 81 drugs were used (the drug list is available for download [84]. A pharmacology expert selected the drugs mainly based on widespread use in the U.S. market. The set also includes relatively newer drugs which were released between 2007 and 2010; this provided a time cushion for market growth and helped to ensure that we could find patient discussions on social media. Major categories include, but not limited to, drugs for the central nervous system and mental health conditions such as Alzheimer's disease and schizophrenia. Treatments for age-related diseases such as diabetes, cardiovascular diseases, urinary dysfunction, and musculoskeletal disorders also met the criteria for potential widespread use, considering increased life expectancy. For more information about the data, and the collection process please refer to prior publications using Twitter data or DS [11,37].

Data Collection

This section provides a summary of the data collection methods from DailyStrength and Twitter that we published in our earlier works [37,38,85].

In DailyStrength every drug has a page with general information about the drug. DS provides the platform for patients or care-givers to share their experiences with others and write reviews about their medications. The reviews for a drug are all listed in the drug web page. We developed a web crawler to collect the user posts about our target drugs.

However Twitter data collection required more processing steps. We used an extended drug list as keywords to monitor the tweets. Besides generic and brand names of drugs, the list also included the misspelled drug names. This was critical to obtaining relevant tweets, as drug names are often misspelled in social media. We generated the misspellings through a phonetic spelling filter [85]. This gave preference to variants that reflect the phonemes of the correct spelling. For example consider the drug name "Seroquel" and the tweets with misspelled drug name:

- @PsychoIogicaI HA! Not if you're on # **Seroquil** . EXTREMELY vivid dreams that stay in conscious memory. Very # Freaky ! Any idea why?
- Gone from 50mg to 150mg of **Serequel** last night. Could barely wake up this morning and I feel like my body is made of lead

Initially, the tool generated a large number of misspellings, out of which 18% were added to the list of drug names. This percentage was experimentally determined to maximize the tweet coverage while minimizing the number of terms needed to query Twitter. This is important because currently Twitter API allows only 400 keywords per application key. This technique allowed us to capture an estimated 50 to 56% of tweets mentioning the drug.

In Table 1, we provide examples of the drug spelling variants in our Twitter API search.

Table 1:Examples of drug spelling variants.

Drug Name	Generated Variants
Prozac	prozaac, prozax, prozaxc
Paxil	paxl, pxil, paxol
Seroquel	seroquels, seroqul, seroqual
Olanzapine	olanzapin, olanzapoine, olanzaoine

We used the publicly available Twitter API to access the tweets that contained the drug names in our list. We could obtain matching tweets up to a volume equal to the streaming cap (~1% of all public tweets), restricted to 1,000 requests per day[3].

Next, we balanced the dataset to select a set of tweets for manual annotation. This helped prevent dominance of some drugs over other drugs, as some drugs are much more popular than others and have a large number of tweets. We randomly selected a maximum of 300-500 tweets per drug, for a total of 10,822 tweets. All twitter datasets were stored and retrieved for later use using a Mongo database [86].

Corpus Annotation

A team of two expert annotators independently annotated the user posts under the supervision of the expert pharmacologist. The annotations include mentions of medical signs and symptoms with the following semantic types:

- adverse drug reaction – a drug reaction that the user considered negative;
- beneficial effect – an unexpected positive reaction to the drug;

[3] https://dev.twitter.com/discussions/4120

- indication – the condition for which the patient is taking the drug; and
- other – any other mention of signs or symptoms.

Every annotation includes the span of the mention (start/end position offsets), the entity type, the related drug name, and the corresponding UMLS CUI (Concept Unique Identifier) — assigned by manually selecting concepts in the ADR lexicon (see next section).

Annotation of the tweets comprised two steps. First the annotators annotated the user posts for binary presence of ADRs. Following that, the tweets with ADRs were separated for full annotation. To have a balanced corpus, the full annotations also included a random set of tweets tagged as not having ADR, and were annotated for other existing mentions (e.g. Indications).

Annotators held weekly meetings to discuss the annotated tweets, correctness of concept labels, and develop annotation guidelines. The two annotators had medical or biological science background. Some meetings also included the full project team (one biomedical informatics student with a computer science background, two computer science students, and a pharmacology doctor).

The annotators annotated a total of 10,822 tweets, utilizing the following general principles for the concept annotation:

- Location boundaries of every mention should be minimized but the boundaries must also capture the entire concept
- Every annotation should be normalized to a UMLS CUI that most *closely* matches the meaning
- For indirect matches, the most *general* ID should be used.

For instance, “weight gain,” “gained 20 pounds,” “put on too much weight,” or “fat fat fat” would all be annotated to a general concept ID for “weight gain.” Instances that caused confusion in selecting the most general term were discussed during meetings. For more information on the annotation process, please refer to the annotation guideline [87]. To measure the inter-annotator agreement, we used *Cohen’s kappa* approach [88]. The calculated kappa value for approximate matching of the concepts is 0.85 for DS and 0.81 for Twitter, which can be considered high agreement [89]. Finally, we generated the gold standard by including only the user posts with complete inter-annotator agreement. From the DS corpus, we randomly selected 4,720 reviews for training (DS train set) and 1,559 for testing (DS test set). The Twitter corpus contains 1,340 tweets for training (Twitter train set) and 444 test tweets (Twitter test set). The Twitter annotated corpus is made available for download [90].

For unsupervised learning, we collected an additional DS user reviews, associated with the most-reviewed drugs in DS, and drug related tweets for a total of more than 3 million sentences. This unlabeled set (Unlabeled_DS_Twitter set), excludes the sentences in DS test and Twitter test sets.

ADR Lexicon

We compiled an exhaustive list of ADR concepts and their corresponding UMLS CUIs. The lexicon, expanded from the earlier work by Robert Leaman [10], and currently includes concepts from COSTART (Coding Symbols for a Thesaurus of Adverse Reaction Terms), SIDER (Side Effect Resource, containing known ADRs) [53,91] and a subset of CHV (Consumer Health Vocabulary, containing consumer alternatives for

medical concepts) [92]. The CHV lexicon contains around 50,000 entries that many of them are not a sign or symptom (possible adverse effect mention), and instead are other health related concepts such as diseases or medical procedures. In order to compile a list of only ADRs, we filtered the CHV phrases by excluding the concepts with UMLS CUIs that were not listed in SIDER. For example, we did not add "West Nile virus" since the related UMLS CUI (C0043125) was not listed in SIDER. The final lexicon contains over 13,591 phrases, with 7,432 unique UMLS CUIs. In addition, we compiled a list of top 136 frequent ADRs tagged by the annotators in the training data. We did not use this additional list during annotation; we only used it in our automatic extraction techniques. The ADR lexicon has been made publicly available [93].

In summary, this chapter presented the details about the collection of user posts from DailyStrength and Twitter. We discussed the challenges and the applied methods in targeting and collecting the tweets about drugs. The information about annotation process and creating the gold standard also presented in this chapter. Finally we explained the details of ADR lexicon which is used during annotation and later for automated concept extraction.

4 METHODS

This section describes our proposed NER systems, ADRMine and DeepHealthMiner, that both utilize deep learning techniques to address the challenges associated with medical concept extraction from social media. The proposed approaches are compared with several baseline extraction techniques including MetaMap, pattern-based concept extraction and a strong baseline lexicon-based approach. The lexicon-based method uses an advanced information retrieval approach for finding candidate lexicon entries in the given text and then utilizes a Support Vector Machine (SVM) classifier to differentiate the possible entity types.

We first explain the details about generating the word embeddings in the following section. ADRMine is presented in Section 4.2 followed by details about DeepHealthMiner in Section 4.3. The baseline techniques are described in Section 4.4.

Section 4.1 Learning the Word Embeddings

One of the main challenges in analyzing social media content is the variety of phrases and sentence structures that people use to express the same or similar meanings. Even if we have a very large training set, there are still many new and creative phrases that are not observed in the training data. Although this is not specific to social media content, it is aggravated in the health domain and for this type of informal text. The conventional NLP systems may struggle with unseen or rarely occurring tokens. This motivated us to utilize the large volume of available unlabeled user posts, by training the word embeddings (see Word Embeddings Section). We used the word embeddings for representing the similarity between the words in our proposed methods. Therefore, in the

case of unseen or rare words, the related classifier could still generalize well, since it was trained on similar words, and the similarity was effectively modeled in the system. For the NER task, we used the word embeddings originally or define classification features based on them (see Embedding cluster features).

We generated 150-dimensional vectors using the word2vec tool [94]. To train the language model, word2vec requires a large input text corpus. We utilized more than 3.5 million user sentences including posts in DailyStrength and Twitter (Unlabeled_DS_Twitter set) to train the word embeddings. In order to choose the embedding's dimension (=150), we performed preliminarily extrinsic evaluations, by measuring the F-measure of the extraction system.

For preprocessing, we split the sentences in every user post, lemmatized all the tokens, and lowercased them for generalization. Furthermore, all user IDs in tweets replaced with the keyword "username" and the digits replaced with letter "d". More details about generating the word embeddings explained in Word Embeddings section.

Characteristics of Health Related Word Embeddings

The generated word embeddings represent interesting semantic characteristics of the words from the input user sentences about drugs. We selected some example words including signs/symptoms, diseases and drugs that were commonly observed in the user posts, and then listed the top similar words to the target word in the embedding space (Table 2). The closeness calculated based on vector cosine similarity scores provided based on Word2vec.

For Instance, from Table 2, consider “Metformin” which is a drug used to treat type 2 diabetes. The top similar word to metformin is “Avandia” which is also used for type 2 diabetes treatment. The second top similar word is “pcos” (Polycystic Ovary Syndrome) which is a disease that Metformin is commonly used to treat that. Similar to Avandia, Glyburide and Glucophage are all antidiabetic drugs and they all have the closest vector to “Metformin”. While the top similar words to “Prozac” (which is commonly used for treating depression) are all antidepressant drugs.

Table 2: Example words from user posts and the top similar words based on unsupervised learned word embeddings. The similarity rankings are based on word2vec cosine similarity scores. From the top ten closest words to the target word, those with similarity higher that 0.65 are listed. The words in the list are lemmatized and also include misspellings.

Metformin	**Prozac**	**Pain**	**Diabetes**	**Depression**	**Nausea**
avandia (0.70)	zoloft (0.89)	spasticity (0.68)	diabetic (0.73)	depressive (0.77)	dizziness (0.85)
pcos (0.69)	paxil (0.87)	neurapathy (0.67)	typed (0.69)	ocd (0.74)	fatigue (0.76)
glyburide (0.68)	lexapro (0.87)	stiffnes (0.67)	diabeti (0.68)	anxiety (0.74)	light-headednes (0.74)
glucophage (0.68)	celexa (0.87)	paresthesia (0.65)	gestational (0.67)	bipolar (0.71)	dizzine (0.74)
clomid (0.68)	effexor (0.86)	numbness/ tingling (0.65)	mellitus (0.67)	o.c.d. (0.71)	diahreah (0.73)
ovulate (0.67)	wellbutrin (0.84)	ciatica (0.65)	insipidus (0.65)	dysthymia (0.70)	diarrhea (0.73)
follistim (0.67)	cymbalta (0.82)	spasum (0.65)		post-partum (0.70)	vomit (0.72)
synthroid (0.67)	lithium (0.75)			zoloft (0.69)	hyperhidrosis (0.72)
conceive (0.66)	aropax (0.757)			hypocondria (0.69)	nausa (0.71)
cytomel (0.65)				depress (0.69)	headache (0.71)

Section 4.2 ADRMine: Sequence Labeling using Word Embedding Clusters

In this section, we explain ADRMine, a machine learning sequence tagger for concept extraction from social media that we introduced in our prior publication [95]. The

effectiveness of various contextual, syntactic and semantic features are explored. We introduce a novel semantic feature based on word clusters, generated from pre-trained word embeddings explained in the previous section.

Model Training

ADRMine uses a supervised sequence labeling CRF classifier for entity tagging in text. CRF is a well-established, high performing classifier for sequence labeling tasks [13,15,16]. We used CRFsuite, the implementation provided by Okazaki [96], as it is fast and provides a simple interface for training/modifying the input features [15,96]. Generating the input CRFsuite train and test files with calculated features for 88,565 tokens in DS train/test sets took about 40 minutes, while building the CRF model and assigning labels for test sentences took about 2 minutes on a PC with a dual core CPU and 10 GB of RAM running Ubuntu operating system.

The CRF classifier attempts to classify individual tokens in sentences. It was trained on labeled mentions of ADRs and indications. Although the focus was to identify the ADR mentions, our preliminary empirical results showed that including indication labels in the model improves the performance of ADR extraction. We also considered the mentions of beneficial effects as indications, since there were limited number of annotated beneficial effects in the corpus, and they are similar to indications. For encoding the concepts' boundaries in ADRMine, we used the IOB (Inside, Outside, Beginning) scheme — where every token can be the beginning, inside, or outside of an entity type. Therefore, it learned to distinguish 5 different labels: *B-ADR, I-ADR, B-Indication, I-Indication* and *Out*.

CRF Features

To represent the classification candidates (the individual tokens), we explored the effectiveness of several features. Here we explain the features that we used in training the machine learning system. We introduce novel semantic features based on word embeddings and also evaluate the effectiveness of contextual sentiment features.

Baseline Features

- Context features: Context is defined with seven features including the current token (t_i), the three preceding (t_{i-3}, t_{i-2}, t_{i-1}), and three following tokens (t_{i+1}, t_{i+2}, t_{i+3}) in the sentence. The preprocessed token strings are values of these features. Preprocessing includes spelling correction and lemmatization. For spelling correction, we utilized Apache Lucene [97] spell checker library which suggests the correct spelling based on an index of English words. The index was generated using the ADR lexicon and a list of common English words from SCOWL (Spell Checker Oriented Word Lists) [98]. For lemmatization, we used the Dragon toolkit [99] lemmatizer which returns the WordNet [100] root of the input word.
- ADR Lexicon: A binary feature that shows whether the current token exists in the ADR lexicon or not.
- POS: Part of speech of the token, which was generated using Stanford parser [101].
- Negation: This feature indicates whether the token is negated or not. We identified the negations by considering grammatical dependency relations between negation words (e.g., no, not, any, cannot and less) and the target token.

We used Stanford parser to generate the grammatical dependencies [101]. The dependencies represent the grammatical relationships with arguments of a relation being the words. The offset of the word in the sentence is also attached to the word in the relation . For instance consider the sentence: "*This drug had no improving effect*" with the following dependency relations:

det(drug-2, This-1)

nsubj(had-3, drug-2)

root(ROOT-0, had-3)

neg(effect-6, no-4)

amod(effect-6, improving-5)

dobj(had-3, effect-6)

Effect is considered as negated since there is a dependency relation that indicates negation between *effect* and *no* (neg(effect-6, no-4)). We also considered a token negated if it occured in a window of two tokens after a negation word. For instance, *improving* in the example sentence is also considered negated [102,103].

Embedding cluster features

One potential problem with the abovementioned features is that the classifier may struggle with unseen or rarely occurring tokens. To address this issue, we incorporated a set of semantic similarity-based features in the system. As explained in Section 4.1 we model the similarity between words by utilizing the unlabeled user posts and training the word embeddings. We then compute clusters of similar words.

We computed the word clusters using Word2vec, performing K-means clustering on the word embeddings. We grouped the words in the corpus into *n* (=150) different clusters, where *n* is a configurable integer number. In Table 3, we provide examples of generated clusters with a subset of words in each cluster. We defined seven features based on the generated clusters. The features include the cluster number for the current token, three preceding and three following tokens. These features add a higher level abstraction to the feature space by assigning the same cluster number to similar tokens. For instance, as Table 3 illustrates, the drug names "*abilify*" and "*adderall* are assigned to the same cluster, which includes only drug names. We selected the value of *n* and the embedding vectors' dimension based on preliminary experiments targeted at optimizing CRF performance for values of *n* between 50 and 500. The generated word embeddings and clusters are made available for download [84]. In Table 4, we show examples of classification candidates and calculated features.

Table 3: Examples of the unsupervised learned clusters with the subsets of the words in each cluster. c_i is an integer between 0 to 149. The "Semantic category" titles are manually assigned and are not used in the system.

Cluster#	Semantic Category	Examples of clustered words
c_1	Drug	abilify, adderall, ambien, ativan, aspirin, citalopram, effexor, paxil, …
c_2	Signs/Symptoms	hangover, headache, rash, hive, …
c_3	Signs/Symptoms	anxiety, depression, disorder, ocd, mania, stabilizer, …
c_4	Drug dosage	1000mg, 100mg, .10, 10mg, 600mg, 0.25, .05, ...
c_5	Treatment	anti-depressant, antidepressant, drug, med, medication, medicine, treat, …
c_6	Family member	brother, dad, daughter, father, husband, mom, mother, son, wife, …
c_7	Date	1992, 2011, 23rd, 8th, april, aug, august, december, …

Table 4: Calculated features for representing examples of CRF classification instances. Sentence: I had the side effect of a bloody noseADR and hated it.

Token	CRF Features	Class
bloody	t_{i-3}=effect; t_{i-2}=of; t_{i-1}=a; t_i=bloody; t_{i+1}=nose; t_{i+2}=and; t_{i+3}=hate; $cluster_{i-3}$=77; $cluster_{i-2}$=49; $cluster_{i-1}$=49; $cluster_i$=147; $cluster_{i+1}$=116; $cluster_{i+2}$=43; $cluster_{i+3}$=51; is_negated=0; is_in_lexicon=1; POS=JJ (Adjective)	B-ADR
nose	t_{i-3}=of; t_{i-2}=a; t_{i-1}= bloody; t_i=nose; t_{i+1}=and; t_{i+2}=hate; t_{i+3}=it; $cluster_{i-3}$=49; $cluster_{i-2}$=49; $cluster_{i-1}$=147; $cluster_i$=116; $cluster_{i+1}$=43; $cluster_{i+2}$=51; $cluster_{i+3}$=85; is_negated=0; is_in_lexicon=1; POS=NN (Noun)	I-ADR
and	t_{i-3}=a; t_{i-2}=bloody; t_{i-1}= nose; t_i=and; t_{i+1}=hate; t_{i+2}=it; t_{i+3}=.; $cluster_{i-3}$=49; $cluster_{i-2}$=147; $cluster_{i-1}$=116; $cluster_i$=43; $cluster_{i+1}$=51; $cluster_{i+2}$=85; $cluster_{i+3}$=101; is_negated=0; is_in_lexicon=0; POS=CC (Coordinating conjunction)	Out

Choosing the Optimal Configurations for Embedding Features

There is no established approach in the literature for identifying the embedding vector size that can be chosen for a specific corpus. Researchers that are currently working with neural networks often choose the configuration settings based on trial and error. We performed extrinsic evaluation of different vector and cluster sizes. We changed the values between 50 to 500 and evaluated the ADR extraction performance. Although the performance did not vary to a large extent, we found that 150 for both vector size and cluster size generated the highest performance. We also repeated part of these experiments by changing the cluster size for Twitter and keeping the embeddings vector size at 150. Figure 6 illustrates the ADR extraction F-measure variations when changing the cluster size. We achieved to the same conclusion that 150 is the best cluster size for our extraction task.

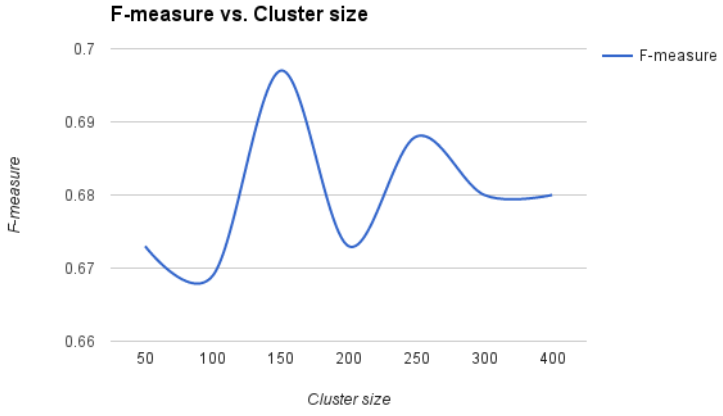


Figure 6: The impact of cluster size on ADR extraction F-measure.

Sentiment Analysis Features

Considering the user posts about drugs, it is easy to see that patients or caregivers usually report the adverse effects with a negative tone, while they report the beneficial and indications in a positive or neutral tone. However capturing the sentiment of the sentences in user posts, and in particular health-related domain introduce some challenges that need to be addressed.

One of the challenges of sentiment analysis in general is that the contextual polarity of the words can be very different with the prior polarities. For example, consider the sentence "*This drug prevents **anxiety** symptoms*", which can be considered a positive sentence; however, the sentence only contains one polar word (*anxiety*) which its prior polarity is negative based on the affect lexicons. Therefore, the existing sentiment analysis methods [104] that are based on the prior polarities do not address the challenges in this task.

Furthermore, a sentence can contain both positive and negative clauses and the contextual polarity can be switched with "Contrastive conjunctions" *("but", "however", "in contrast", "on the contrary", "instead", "nevertheless", "yet", "still", "even so", "neither ... nor"*). For instance, consider the importance of contrastive conjunctions in switching the contextual polarity in the following sentences:

1) *[Wonderful]*$^{+}$ *but [stopped taking it because of weight gain]*$^{-}$.
2) *[Made me forget things]*$^{-}$ *but [slept like a rock which was great when manic]*$^{+}$.

To model the contextual polarity of a token, we defined two sentiment-related features: token contextual polarity and sentence polarity. The sentence polarity feature is the

overall polarity of the sentence and can have three possible values: positive, negative and neutral. Similarly, token contextual polarity feature can have three possible values. For calculating the value of this feature , our method does not consider the tokens in isolation and instead considers the sequence of tokens in the sentence. For example, if the token is preceded by a contrastive conjunction, or if it is negated, its polarity may be switched from positive to negative or vice versa. We used Stanford CoreNLP (version 3.3.0) to parse a given sentence and assign sentiment to the subtrees in the sentence parse tree [105]. The tool calculates the polarity values based on a deep learning recursive neural network that is trained to assign phrase-level polarity scores in a sentence. It considers the sequence of words and the composed phrases to assign sentiment values. The tool assigns sentiment to each node in parse tree of a given sentence. Therefore, it is possible to get the sentiment of tokens and phrases at several levels in the parse tree.

To get the contextual polarity for a token, our method first identified the token's related clause based on the parse tree. To identify the related clause of a token (a leaf node in a parse tree), for every leaf node, our technique recursively visited the ancestors until it reached a node with clause level labels (e.g. S (simple declarative clause) or SBAR (Clause introduced by a subordinating conjunction)) [106]. We then used the sentiment of the related clause as the value of the token contextual polarity feature.

Section 4.3 DeepHealthMiner: Deep Learning for Health Information Extraction

In this section we present DeepHealthMiner, a deep learning based pipeline for extraction of health-related concepts. The system learns both the mentions' spans (the start and the

end tokens) and the entity types. Figure 7 shows DeepHealthMiner processing pipeline that incorporates three main steps:

1. Learning the word embeddings
2. Concept extraction
3. Concept normalization

The system used around 3 million unlabeled user sentences for generating the word embeddings using unsupervised learning (see Section 4.1 Learning the Word Embeddings). It utilized these embeddings for representing the individual tokens for the main task of concept extraction. The concept extraction module used a feedforward neural network to learn to tag the medical concepts in the input sentences. In the next section, we explain the neural network sequence classifier structure.

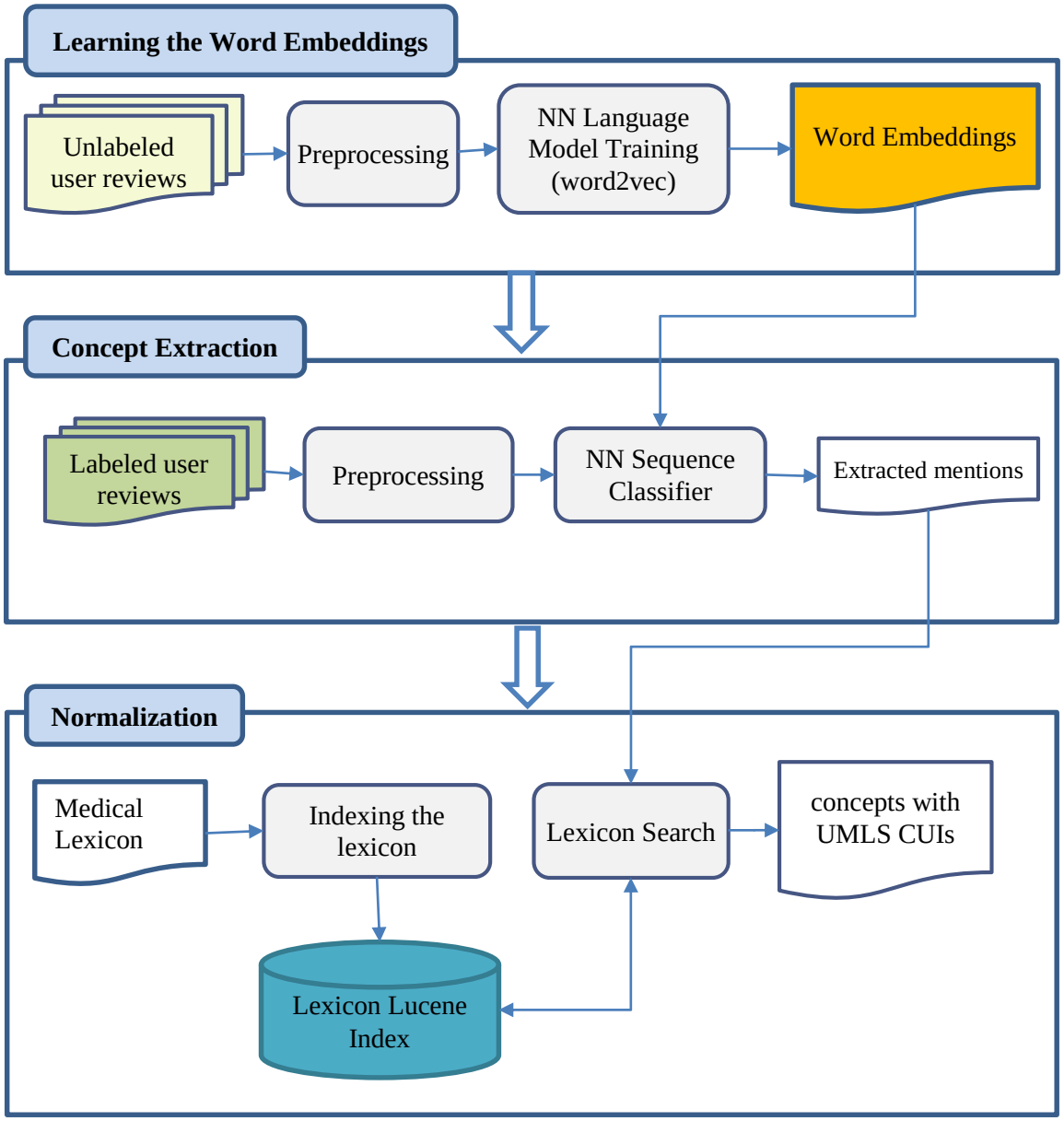


Figure 7: DeepHealthMiner Pipeline.

Deep Neural Network Sequence Classifier

We designed a feedforward neural network classifier (Figure 8) to learn the token labels. The target entities in the training data are medical sign and symptoms found in user posts about drugs. More specifically, the system learned to extract and distinguish ADRs from

indications. Every mention can contain one or multiple tokens. Similar to ADRMine, to represent the annotation boundaries to the sequence classifier, we chose the IOB scheme. As a result, there were five possible classes for every token (B-ADR, I-ADR, B-Indication, I-Indication and O). The outside tag (O) used for the tokens that did not represent a named entity. Following is an example of an annotated sentence using IOB encoding:

Gave	me	**electric**	**shocks**	and	caused	me	to	**gain**	**Almost**	**9**	**POUNDS**	in	3	WEEKS
O	O	B-ADR	I-ADR	O	O	O	O	B-ADR	I-ADR	I-ADR	I-ADR	O	O	O

The network architecture is similar to SENNA, the general purpose neural network suggested by Collobert et al [20] that has shown successful when applied for a number NLP tasks. We used DeepNL [107] a user-friendly implementation of SENNA that can be trained on a given annotated corpus. As illustrated in Figure 8, the network has an embedding look up layer, an input, a hidden and an output layer. It is a fully connected NN, meaning that every node in a layer is connected to all the nodes in the next layer. As Figure 8 illustrates, the input tokens are represented as one-hot first and the look up table retrieves the pre-trained word embeddings for every token. DeepHealthMiner uses a window approach [20] in which it concatenates the input embedding vectors and pass to the input nodes. Therefore, the size of the input layer equals the embedding dimension (here is set to 150) times the window size (here is set to 7).

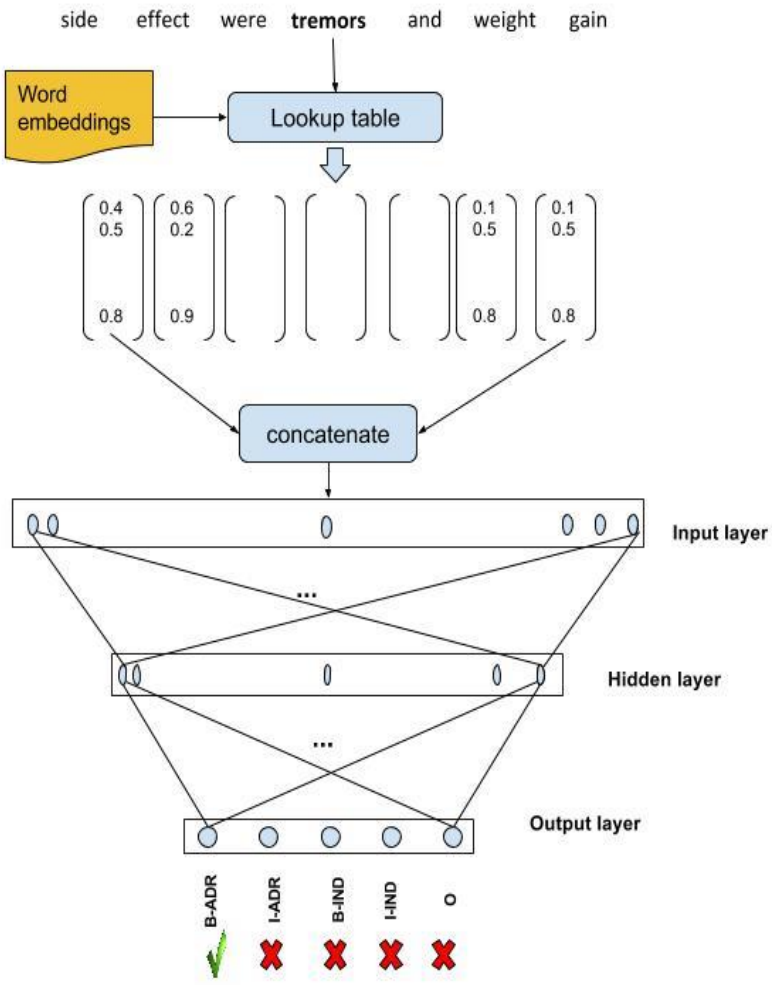


Figure 8: Neural network sequence classifier architecture.

A training candidate is a pair denoted as $(x^{(t)}, y^{(t)})$, where $x^{(t)}$ is the target token and the context tokens around it, and $y^{(t)}$ represents the label for the target token. For instance, Equation 5 models the input to the network for a token, x^t, with a context window of 5:

$$x^{(t)} = [x_{t-2}L, x_{t-1}L, x_tL, x_{t+1}L, x_{t+2}L\ ]$$

Equation 5: The context window including target token and the neighbor tokens.

Where:

- $x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}$ are one-hot vectors of size $|V|$ equal to the vocabulary size. The index of the word in the vocabulary is set to one and the rest of dimensions are 0.

$L \in R^{|V| \times d}$ is the embedding matrix with rows containing the embedding vectors for words in the vocabulary. The i^{th} row in L is an embedding vector corresponding to the i^{th} word in the vocabulary. d is the embedding dimensionality which is set to 150 in this study.

For a sentence with *n* tokens, we have *n* separate training examples corresponding to every token. In the training data, if a single token is labeled as ADR, the expected output for that instance is B-ADR, and the output node that represents B-ADR is set to "1" and the rest of the outputs are set to "0". This is represented as (1,0,0,0,0), while if the token is labeled as indication (B-IND), it is represented as (0,0,1,0,0).

The NN adjusts the weights connecting the input-to-hidden and hidden-to-output weights. Hidden layer generates a higher level representation of the input in a way that it makes the classification task easier for the output layer. The hidden layer contains sigmoid neurons which use hyperbolic tangent function to add non-linearity to the weighted sum of inputs (see Equation 2).

The goal of the network is to adjust the internal parameters in a way that the right label is assigned to each word. Similar to other neural network-based classification tasks [20,81], the network is trained using standard backpropagation to maximize the probability of the class (y_t) of the input word (x_t):

$$y_t = \arg\max p(y_t | x_{t-2} \dots x_{t+2})$$

Equation 6: Predicted label for x_t.

Following the convention [20,75], the network uses cross entropy (Equation 4) as the loss function during training. After training, the output represents the probability distribution of the class of the input word, given the context of the word. The output is a softmax layer that is typically used for classification tasks (see Deep Learning techniques for NER).

Network Parameter Selection

Neural networks can automatically extract classification features, but to achieve the optimal performance of the networks we have to first choose the neural network architecture and also set several hyper-parameters of the neural network. For instance, the input window size, the number of hidden layers, number of hidden nodes at each hidden layer, and the learning rate are examples of the parameters that should be selected.
To choose the network parameters, we changed one parameter at a time while keeping others constant. Based on empirical results we designed a single hidden layer NN classifier with an input window size of 7 (including the target token, three preceding and three following tokens) and an output softmax layer with five nodes (see DeepHealthMiner Parameter Selection section).

The Impact of Deep Learning on Required Train Set Size

One of the challenges for training supervised machine learning systems is the need to manually annotate training instances. The annotation effort is very costly and cumbersome. The annotation of health-related user posts in social media, and in

particular Twitter, is even more challenging since the user posts are more informal which leads in dealing with more ambiguity (in deciding about the mention span, semantic type or mapping UMLS concept IDs) during annotations.

To systematically test the impact of deep learning on the size of the training data, we trained the system on smaller data sets while keeping the test set intact. First we randomly selected 10% of the train set, trained ADRMine, DeepHealthMiner and the baseline CRF and compared the results by testing them on the whole test set. Next we increased the train set size to 25% and 50% of the size of the training set and compared the results accordingly. Since we have a relatively smaller corpus on Twitter, we only compared 50% of the Twitter corpus with the whole train set.

Normalization

After extracting the medical entities and the entity types, we mapped the extracted concepts to the concept IDs using UMLS. In Table 5, we list some example mentions and the associated UMLS CUIs from our annotated corpus. For instance, the normalization system should map "*nightmare*" to C0028084; Considering the associated UMLS concept name, which is the same as the extracted mention (*nightmares*), the normalization task for this example is relatively easy. However, normalization of mentions that do not share tokens with the associated UMLS concept names is more

Table 5: Examples of annotated mentions and the normalized UMLS CUIs.

Sentence	Entity Type	UMLS CUI	UMLS Concept Name
I swear this [DRUG_NAME] is causing me to have horrible **nightmare**!	ADR	C0028084	nightmares
@[username] I take [DRUG_NAME] and have a permanent **dry mouth** what can I do to help it?	ADR	C0043352	mouth dryness
stops me from crying most of the time, **blocks most of my feelings**	ADR	C0233469	emotional indifference
YES, one of the best things to help with my **mood stabilization**.	Indication	C0085633	mood swings

challenging. For example, "**blocks most of my feelings"** should be mapped to C0233469 (emotional indifference).

Here we present a preliminary approach that used the ADR Lexicon for normalization. The method is similar to our baseline technique (see Lexicon-based Concept Extraction). The system first indexes all the lexicon entries, including the UMLS CUIs, using Lucene [97]. Each lexicon entry is indexed as a Lucene document that can be retrieved later.

To normalize the extracted entities, the method generates a query using the extracted span of text and it retrieves a ranked list of all the lexicon entries that contain full or part of the included tokens in the query. It then selects the top ranked UMLS CUI based on a chosen threshold. In the Lexicon-based Concept Extraction section we provide more details.

Section 4.4 Baseline Extraction Techniques

Lexicon-based Concept Extraction

To locate the ADR lexicon concepts in user sentences, we used an information retrieval approach based on Lucene, which is similar to those applied for ontology mapping [108,109] and entity normalization [110]. We built a Lucene index from the ADR lexicon entries. For each concept in the lexicon, we added the content and the associated UMLS CUI to the index. Before indexing, the concepts were preprocessed by removing the stop words and lemmatization.

To find the concepts presented in a given sentence, we generated a Lucene search query after preprocessing and tokenizing the sentence. The retrieval engine returns a ranked list of all the lexicon concepts that contain a subset of the tokens presented in the input query. We considered a retrieved concept present in the sentence if all of the concept's tokens are present in the sentence. We then used string comparison via regular expressions to identify the span of the mentions in the sentence. This technique is flexible enough to identify both single and multi-token concepts, regardless of the order or the presence of other tokens in between them. For example, the sentence "*... I **gained** an excessive amount of **weight** during six months.*" is correctly matched with the lexicon concept "*weight gain*". We applied two constraints before accepting the presence of a retrieved lexicon concept: the distance between the first and the last included token should be equal or less than a configurable size (= 5), and there should not be any punctuation or connectors like '*but'* or '*and'* in between the tokens.

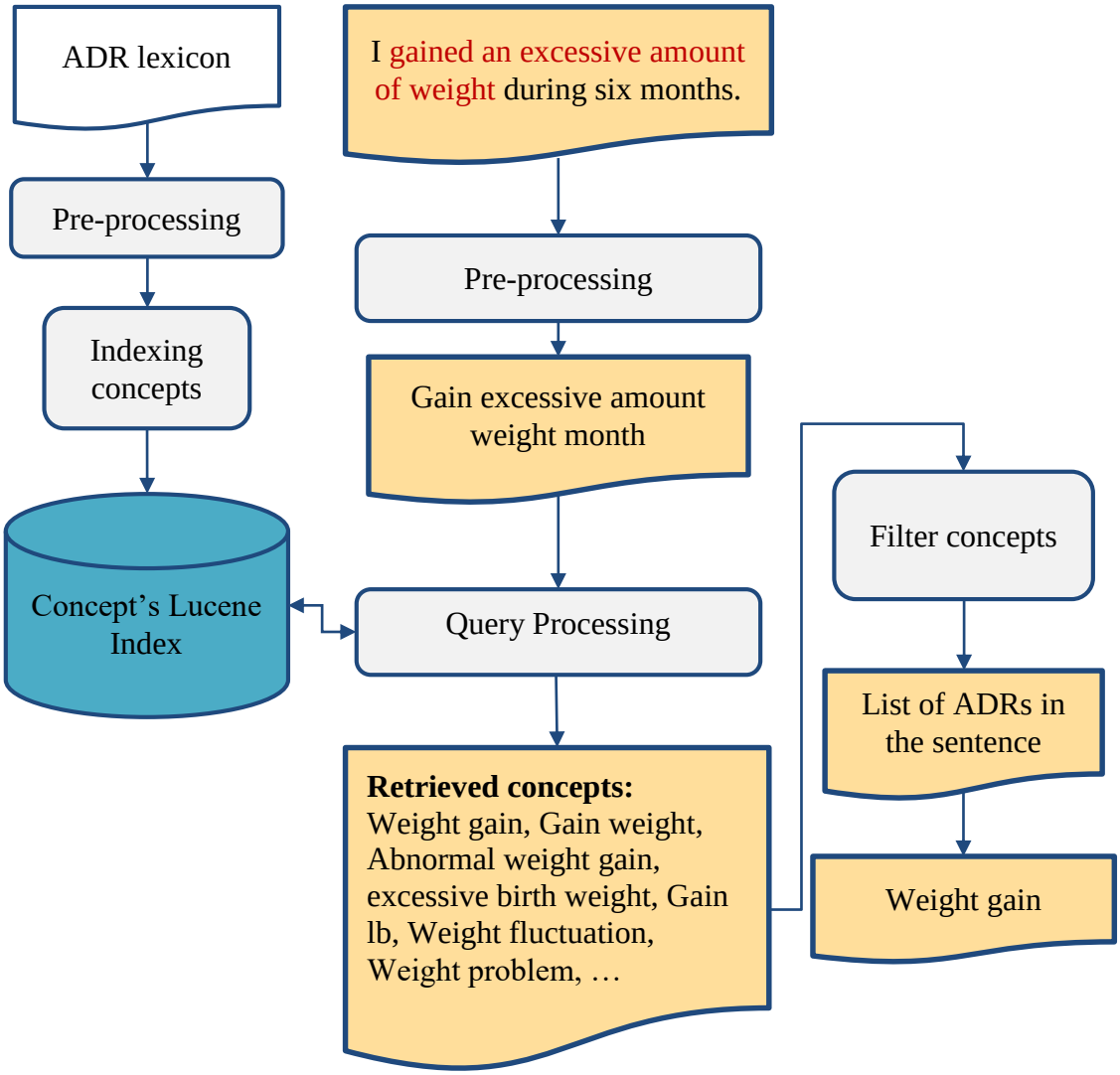


Figure 9: Lexicon-based concept extraction method pipeline.

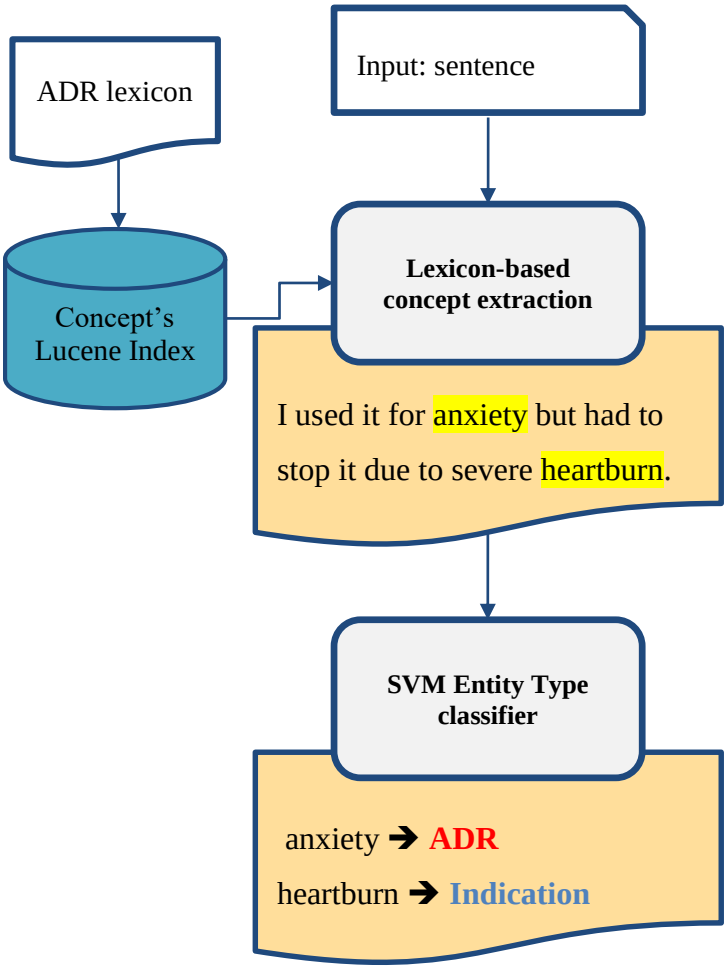


Figure 10: Entity type classification pipeline.

SVM Entity Type Classifier

Since not all mentions that match with the lexicon are adverse reactions, we trained a multiclass SVM classifier to identify the entity types of the candidate phrases. Every SVM classification candidate is a phrase (may include more than one token) that is already matched with the ADR lexicon. The possible entity types for a candidate phrase

are *ADR*, *Indication* or *Other*. We chose SVM because it has been shown to perform very well in text classification problems [111]. We used SVM^{light} [112] to build the SVM model. The SVM features for representing the candidate phrases are similar to CRF features and include: the phrase tokens, three preceding and three following tokens around the phrase neighbor tokens), the negation feature, and the embedding cluster number for the phrase tokens and the neighbor tokens.

Pattern Mining for ADR Extraction

In this section we introduce a new method to automatically extracting ADRs from user comments using natural language processing techniques that go beyond lexicon matching. We applied association rule mining, a supervised learning method, to extract mentions of ADRs in user reviews about drugs in social media. The hypothesis that drives our method is that even if the language used in social media is highly informal, people write their comments using some converging patterns that can be identified to facilitate the extraction of interesting pieces of information in those comments.

Association Rule Mining

The idea of association rule mining originated from the "shopping cart" problem, where the challenge is to identify which set of items are more likely to be bought together. Supermarkets use this information in positioning the items in the shelves and controlling the way customers traverse in the supermarket. Association rules are represented as a set of expressions of the form $\{X_1,X_2,X_3,..X_n\} => Y$, which indicates that if we find $X_1,X_2,X3,\ldots X_n$ in a shopping cart (a transaction), the probability of finding another product Y in that transaction will be high. This probability is called the *confidence* of the

rule, and usually one seeks the rules with confidence above a defined threshold. In addition, the number of transactions that include all the items X_1... X_n and Y together is called the *support* of the rule. A frequent itemset is a set of items which have the support and confidence higher than a defined threshold. The Apriori algorithm [113] is an influential algorithm in mining frequent association rules. It iteratively traverses transactions to find itemsets with cardinality from 1 to K (K-items) [114]. The dominant rule behind Apriori method is that if S is a frequent itemset, every subset of S should be frequent also. Once the frequent itemsets are found, they are used to generate the rules.

Mining patterns in the free text can be modeled as an association rule mining problem in which every sentence is considered a transaction and the words in the sentence are considered as items in the transaction.

Concept Extraction with Association Pattern Mining

The proposed pattern-based concept extraction technique [11] is based on three main steps: 1. Term Sequence Generation, 2. Frequent Rule Identification and 3. Frequent Pattern Generation. In Figure 11, we illustrate the main steps in processing the user posts.

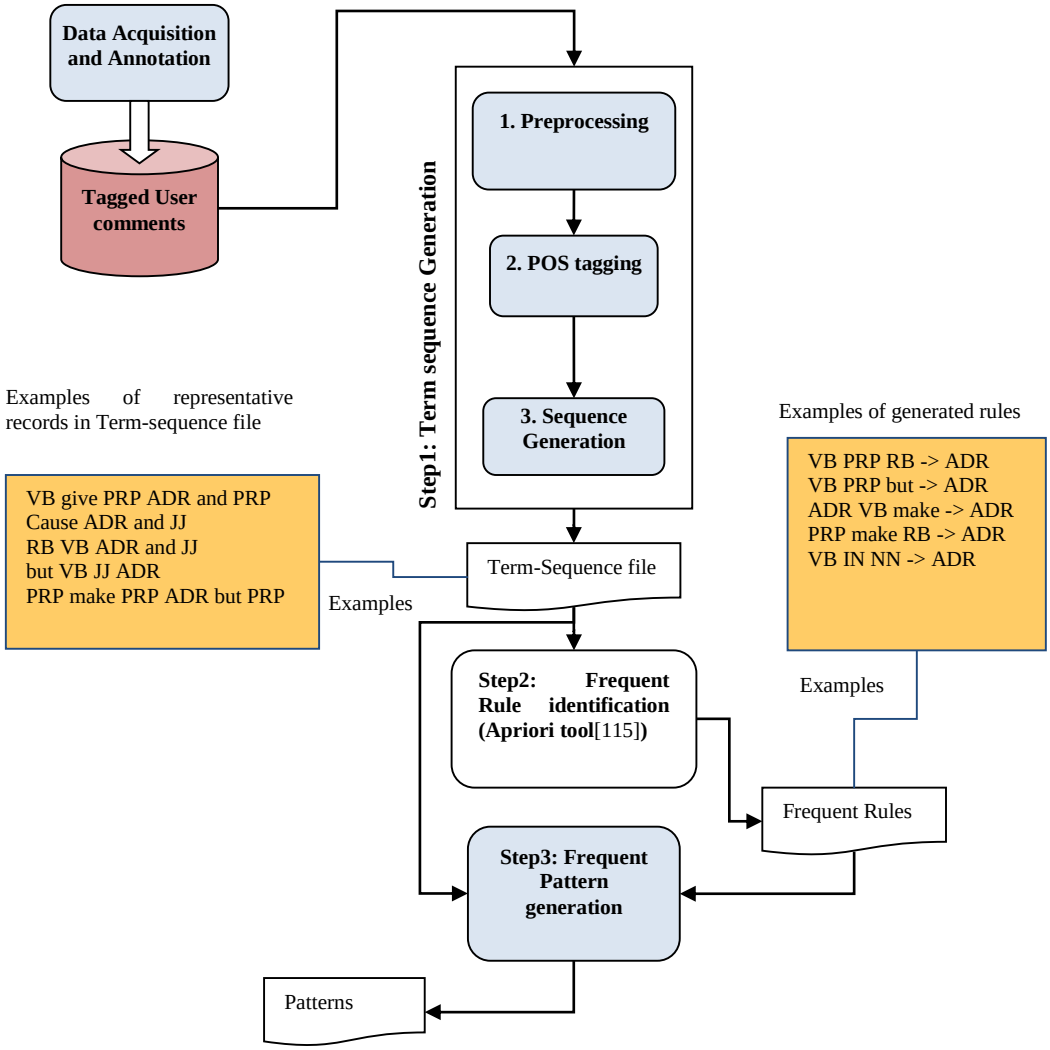


Figure 11: Overall architecture of the pattern-based concept extraction system.

Step 1: Term Sequence Generation

First, we created a collection of sequences of words that appeared in sentences with an ADR mention, and stored them in a "Term-Sequence" file. Each line of this file is thus representative of a sentence with an ADR mention. Either a generic part of speech or the original words in the sentence constitute the sequence elements which we refer to as

terms. To generate the representative Term-Sequences for a labeled sentence in the training data , we selected a few terms from a window around each side of the ADR mention. We used part of speech as representatives for all words except for connectors (such as “and”, ”but”) and a few manually selected verbs such as (make, cause, give).

Step 2: Frequent Rule Identification

At this step, we extracted a set of rules that tell us which combination of terms are more likely to be present in a sentence with a mention of an ADR. The Term-Sequence file generated in the previous step is the input to this module. We applied Apriori algorithm for association rule mining [113], using the Borgelt's implementation [115]. For instance, consider the following extracted rule: “make PRP RB CC => ADR” which means that the combination of the verb (“make”), a preposition (PRP), adverb (RB), and a connector (CC) often occurs with an ADR (in no particular order). This allows us to infer that if the combination in the condition is present in the test sentence, a mention of an ADR is likely to be present, as well.

Step 3: Frequent Pattern Generation

We generated the sets of frequent terms using the rules from the previous step; however, the order of these terms should be identified to generate the patterns. We used the Term-Sequence file, which includes all the possible sequences, to find the orders. ADR keywords in the sequence are replaced with a placeholder that catches the term(s) presented in that position in the test sentence. In Table 6, we present an example of the way that the rule is converted to some of the possible patterns.

Table 6: Converting a rule to possible patterns

Rule	Possible Sequence	Patterns
PRP and make -> ADR	make PRP ADR and	make PRP (.*) and
	and make PRP ADR	and make PRP (.*)
	make PRP JJ and ADR	make PRP (?:[^]*)*and (.*)
VB RB and -> ADR	VB ADR RB and	VB (.*) RB and
	RB ADR and VB	RB (.*) and VB

Testing Phase: Extracting ADRs from unseen comments

We generalized every test sentence and generated the representative sequence, following the same approach explained in step1 (Term-Sequence generation). Note that, for testing, we do not have an ADR keyword in the representative term sequence, and the goal is to extract them by applying the patterns. We applied all the patterns generated in Step3 to the test sentences to extract adverse effect mentions. In Table 7, we show a pattern and examples of matched sentences.

Table 7: Example of test sentences that match with a given pattern

pattern	Example sentences	Extracted ADR
cause PRP (.*)	It works, but swell too much and get heart murmurs	swell too much
	I think it actually causes me have more headaches	have more headaches

MetaMap Baselines

We used MetaMap to identify the UMLS concept IDs and entity types in the user reviews, and add two baselines to evaluate the performance of MetaMap on this type of data. In the first baseline ($\text{MetaMap}_{\text{ADR\_LEXICON}}$), all identified mentions by MetaMap that their assigned UMLS CUIs are in our lexicon are considered to be ADRs. In the second

baseline ($\text{MetaMap}_{\text{SEMANTIC\_TYPE}}$), all concepts belonging to specific UMLS semantic types are considered to be ADRs. The selected semantic types include: injury or poisoning, pathologic function, cell or molecular dysfunction, disease or syndrome, experimental model of disease, finding, mental or behavioral dysfunction, neoplastic process, signs or symptoms, mental process.

This chapter presented the proposed methods for medical concept extraction from social media. The details about learning the word embeddings from unlabeled user posts and the characteristics of the learned embeddings presented in the first section. It then was followed by a section about ADRMine – the CRF sequence tagger- and the prosed embedding cluster features. We then presented DeepHealthMiner, our deep learning pipeline for concept extraction. Finally the information about baseline NER techniques were presented in detail. Next, we present the evaluation results and compare the performance of several concept extraction techniques.

5 EVALUATION AND RESULTS

We evaluated the performance of the extraction techniques using precision (p), recall (r) and F-measure (f):

$$p = \frac{tp}{tp+fp} \qquad r = \frac{tp}{tp+fn} \qquad f = \frac{2*p*r}{p+r}$$

We calculated true positives (tp), false positives (fp) and false negatives (fn) by comparing the systems' extracted concepts with the manually tagged concepts in the gold standard via approximate matching [116]. To evaluate the effectiveness of the proposed techniques we used two different corpora: DailyStrength (DS) and Twitter. In Table 8, we present the details about the sentences and the number of annotated concepts in each corpus. The annotated Twitter data set is available for download [90].

We also used the data released for PSB 2016 Social Media Mining Shared Task [117] to evaluate the performance of ADRMine and DeepHealthMiner. We have made this data set accessible for download which can be helpful to the future research.

Table 8: Number of user posts and annotation details in train/test sets.

Data Set	# of user posts	# of sentences	# of tokens	# of ADR mentions	# of Indication mentions
DS train set	4,720	6,676	66,728	2,193	1,532
DS test set	1,559	2,166	22,147	750	454
Twitter train set	1,340	2,434	28,706	845	117
Twitter test set	444	813	9,526	277	41
PSB train set	1784	3247	38232	1122	158
PSB test set	476	995	11266	574	275

Section 5.1 ADRMine Evaluation

In Table 9, we compare the performance of ADRMine with the baseline techniques. We found that ADRMine significantly outperforms all baseline approaches (p-value < 0.05). Furthermore, the utility of different techniques in concept extraction is consistent between the two tested corpora. We computed the statistical significance (p-value) by using the model proposed by Yeh [118] and implemented by Pado [119].

Table 9: Comparison of ADRMine and the baseline methods. The ADR extraction precision (P), recall (R) and F-measure (F) are compared using two different corpora: DS and Twitter.

	DS			Twitter		
Method	**P**	**R**	**F**	**P**	**R**	**F**
$MetaMap_{ADR\_LEXICON}$	0.470	0.392	0.428	0.394	0.309	0.347
$MetaMap_{SEMANTIC\_TYPE}$	0.289	0.484	0.362	0.230	0.403	0.293
Pattern Mining	0.775	0.475	0.589	0.546	0.126	0.205
Lexicon-based	0.577	0.724	0.642	0.561	0.610	0.585
SVM	0.869	0.671	0.760	0.778	0.495	0.605
Baseline CRF	**0.874**	0.723	0.791	**0.788**	0.549	0.647
ADRMine	0.860	**0.784**	**0.821**	0.765	**0.682**	**0.721**

Evaluation of CRF Features

To investigate the contribution of each feature set in ADRMine, we performed leave-one-out feature experiments (Table 10). We found that the most contributing groups of features are the context (see *Baseline Features*) and the embedding clusters. The

combination of both is sufficient to achieve the highest result for DS. In Table 11, we report the evaluation of sentiment features when added to ADRMine.

Table 10: The effectiveness of different CRF feature groups. All feature set (All) includes: context, lexicon, POS, negation and embedding clusters (cluster). Statistically significant changes ($p<0.05$), when compared with All feature set, are marked with *.

CRF Features	DS			Twitter		
	P	**R**	**F**	**P**	**R**	**F**
All	0.856	0.776	0.814	0.765	**0.682**	**0.721**
All – lexicon	0.852	0.781	0.815	0.765	0.646	0.701
All – POS	0.853	0.776	0.812	0.754	0.653	0.700
All – negation	0.854	0.769	0.810	0.752	0.646	0.695*
All – context	0.811	0.665	0.731*	0.624	0.498	0.554*
All – cluster	0.851	0.745	0.794*	**0.788**	0.549	0.647*
context + cluster	**0.860**	**0.784**	**0.821***	0.746	0.628	0.682*

Table 11: Evaluation of sentiment features. ADRMine for DS includes context and embedding cluster features, and for Twitter includes context, lexicon, POS, negation and embedding cluster features.

Features	DS			Twitter		
	P	**R**	**F**	**P**	**R**	**F**
ADRMine	0.860	0.784	0.821	**0.765**	**0.682**	**0.721**
ADRMine + sentiment	**0.861**	**0.803**	**0.831**	0.759	0.650	0.700

To further investigate the power of the embedding clusters, we performed several experiments for comparing them with baseline features. These experiments were only performed on DS as we had a relatively larger set of training data available. We varied

the size of the training data while keeping the test set unchanged. Starting with 20% (944 reviews) of the original DS training set, we increased its size by 20% each time via random sampling without replacement. Figure 12 shows that adding the cluster features (context + clusters) constantly improves F-measure (Figure 12-b), gives significant rise to the recall (Figure 12-a), but slightly decreases the precision.

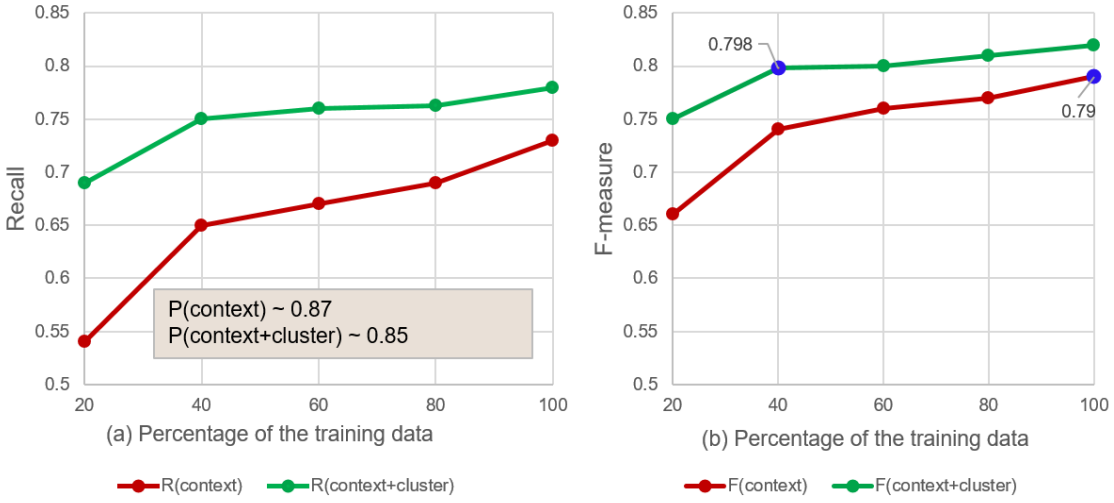


Figure 12: The impact of embedding clusters on precision, recall (a) and F-measure (b). The CRF trained on variable training set sizes and tested on the same test set.

Discussion

We found that ADRMine is capable of extracting complex medical concepts even those that were not seen in the training data or found in the medical lexicons. In Figure 13, we show examples of successfully extracted ADRs and indications using ADRMine. The results indicate that when we have a large unlabeled corpus for generating the embedding clusters, ADRMine can learn from a relatively small labeled data set. The utilized features are general and independent of the entity types or the domain, therefore the

system can easily be trained to extract other types of concepts in medical or other domains.

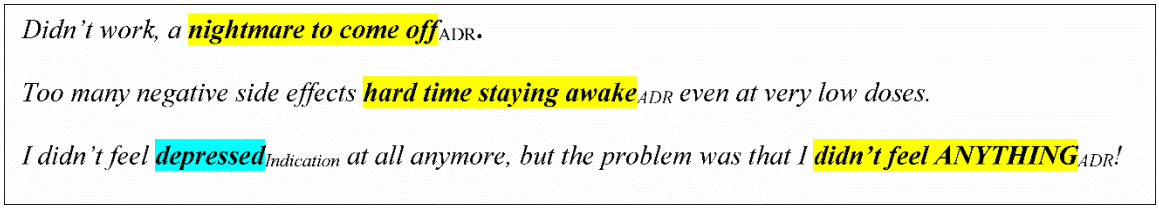


Figure 13: Examples of successfully extracted concepts using ADRMine.

The Effectiveness of Classification Features

Feature evaluations (Table 10) indicated that lexicon, POS, and negation features added no significant contribution to the results when CRF was trained on comparatively larger number of training instances (DS train set), while they could still make small contributions to the performance when less data was available (Twitter train set or DS with less number of training instances).

However, for both corpora, the context features were fundamental for achieving both high precision and recall; and the embedding cluster features were critical in improving the recall which resulted in a significant boost in F-measure. Examples of ADRs that were extracted after adding the cluster features are starred (*) in Figure 14.

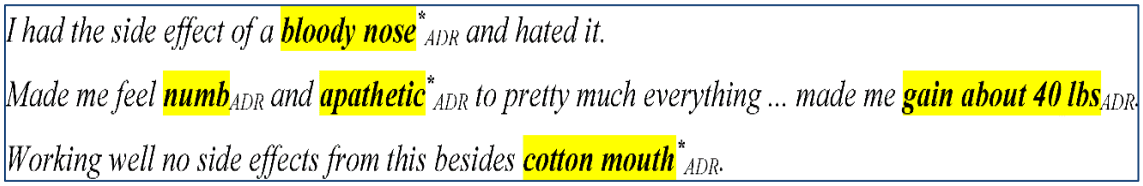


Figure 14: Examples of concepts that could only be extracted after adding the embedding cluster features to ADRMine. These concepts are starred and other extracted concepts are just highlighted.

As Figure 12-b illustrates, the system that used the word embedding cluster features constantly achieved remarkably higher F-measure compared to the system that only used

the baseline features. Interestingly, the F-measure of the CRF with cluster features when using 40% of the training data is even higher than the F-measure when using 100% of the training data but without cluster features (Figure 12-b). Therefore, these features can be more advantageous in situations where less annotated data is available. As shown in Table 9, the contribution of the cluster features in improving the F-measure was substantially higher for the Twitter corpus which also confirms this finding.

We initially anticipated that sentiment analysis would improve the performance of concept extraction. Although adding the sentiment features improved the performance on DailyStrength, surprisingly, they slightly worsen the ADRMine performance on Twitter corpus. The possible reason is related to the nature of tweets which are short and very noisy that heavily deviate from grammatical rules. As a result, the utilized sentiment analysis tool that is trained on movie reviews and is not specifically trained on health-related user posts on Twitter, possibly generates erroneous parse trees on tweets. Furthermore, although movie reviews and user posts about drugs share common characteristics in expressing positive and negative views in general, but health-related posts are more challenging for sentiment analysis. For instance, consider the prior example "*This drug prevents **anxiety** symptoms*"; The polarity of the sentence is positive considering the meaning of "*prevent*" in health domain. However, the Stanford sentiment score that is based on a model trained on movie reviews classifies this sentence as neutral. It is anticipated that training the sentiment classifier on a health-related corpus, annotated for phrase level sentiments, may improve the quality of the assigned sentiment scores and consequently the concept extraction performance.

ADRMine Error Analysis

For error analysis, we randomly selected 50 false positive and 50 false negative ADR mentions from DS test set and categorized the likely sources of errors. In Figure 15, we provide a summary of this evaluation, with example false positive/negative concepts shown within brackets. The majority of false positive errors were caused by mentions that were confused with indications or non-ADR clinical mentions. We believe that incorporating more context (*e.g.*, a longer window) will diminish such errors in future. Twenty eight percent of false negative ADRs were expressed in long, descriptive phrases, which rarely included any technical terms. Sentence simplification techniques might be effective in extracting such false negatives [120]. Irrelevant immediate context or the lack of context in too short, incomplete sentences, made it difficult for ADRMine to generalize, and contributed to 26% of false negatives. Other false negatives were related to specific rules in the annotation guideline, mentions expressed with complex idiomatic expressions, or uncorrected spelling errors. Future research is needed to identify an optimized set of features that could potentially minimize these errors.

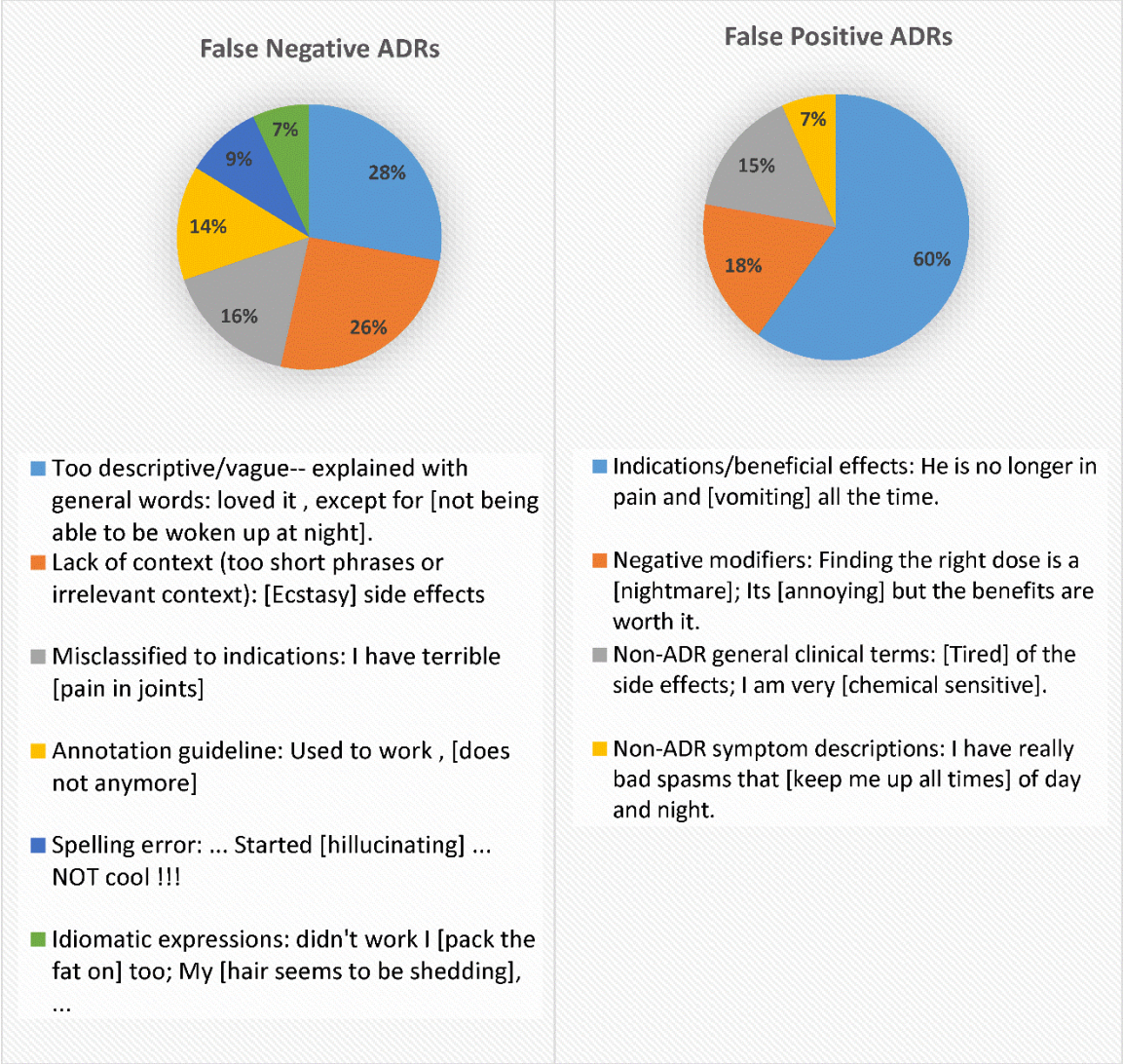


Figure 15: Analysis of ADRMine false positive and false negatives.

Section 5.2 DeepHealthMiner Evaluation

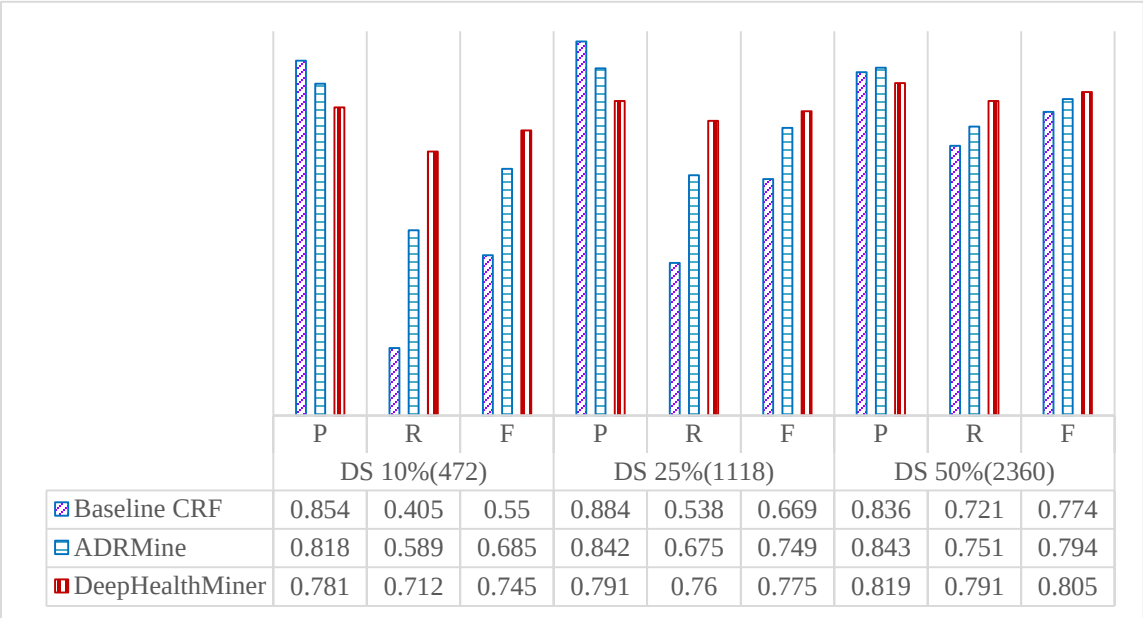
DeepHealthMiner achieved the F-measure of 0.84 on DS and 0.73 on Twitter corpus, outperforming ADRMine and other baseline methods (Table 12). The good performance of DeepHealthMiner is mainly attributed to improving the recall compared to ADRMine and other extraction methods (Table 12). We also evaluated DeepHealthMiner and ADRMine on the PSB shared task 2016 data set and similarly they both significantly outperformed the baseline CRF (Table 13).

Table 12:Comparison of DeepHealthMiner with baseline extraction methods.

	DS			Twitter		
Method	**P**	**R**	**F**	**P**	**R**	**F**
$MetaMap_{ADR\_LEXICON}$	0.470	0.392	0.428	0.394	0.309	0.347
$MetaMap_{SEMANTIC\_TYPE}$	0.289	0.484	0.362	0.230	0.403	0.293
Pattern Mining	0.775	0.475	0.589	0.546	0.126	0.205
Lexicon-based	0.577	0.724	0.642	0.561	0.610	0.585
Lexicon-based + SVM	0.869	0.671	0.760	0.778	0.495	0.605
Baseline CRF	**0.874**	0.723	0.791	**0.788**	0.549	0.647
ADRMine	0.860	0.784	0.821	0.765	0.682	0.721
DeepHealthMiner	0.866	**0.809**	**0.837**	0.768	**0.704**	**0.734**

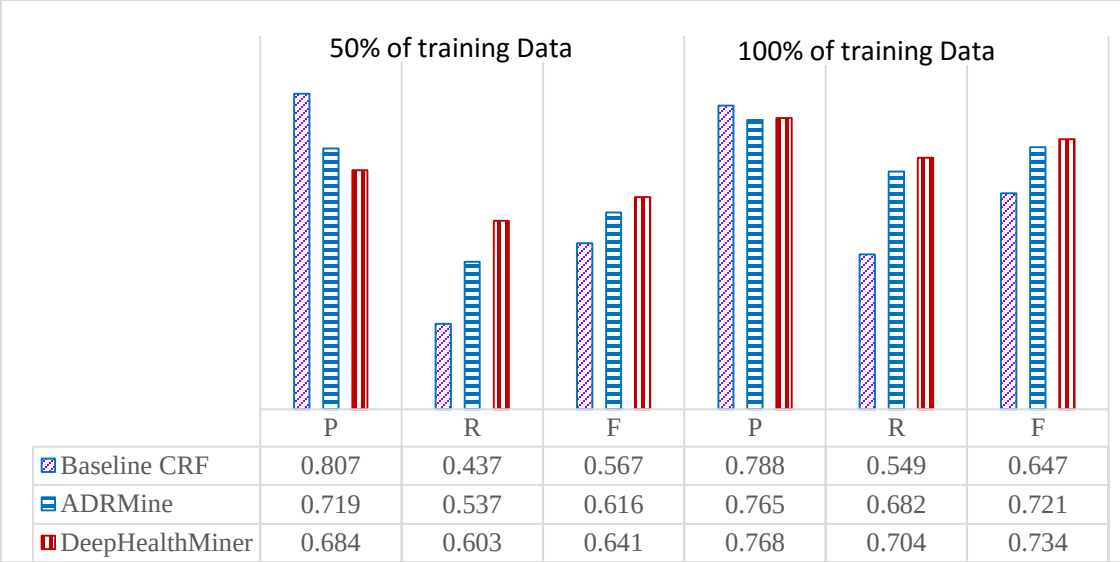
Table 13: PSB shared task 2016 evaluation results.

Method	PSB Twitter Data Set		
	P	**R**	**F**
Baseline CRF	**0.770**	0.462	0.577
ADRMine	0.756	0.545	0.634
DeepHealthMiner	0.718	**0.604**	**0.656**



	P	R	F	P	R	F	P	R	F
	DS 10%(472)			DS 25%(1118)			DS 50%(2360)		
Baseline CRF	0.854	0.405	0.55	0.884	0.538	0.669	0.836	0.721	0.774
ADRMine	0.818	0.589	0.685	0.842	0.675	0.749	0.843	0.751	0.794
DeepHealthMiner	0.781	0.712	0.745	0.791	0.76	0.775	0.819	0.791	0.805

Figure 16: Impact of training set size on performance (precision (P), recall (R) and F-measure (F)) of different extraction methods (DS corpus).



	50% of training Data			100% of training Data		
	P	R	F	P	R	F
Baseline CRF	0.807	0.437	0.567	0.788	0.549	0.647
ADRMine	0.719	0.537	0.616	0.765	0.682	0.721
DeepHealthMiner	0.684	0.603	0.641	0.768	0.704	0.734

Figure 17: Impact of training set size on performance (precision (P), recall (R) and F-measure (F)) of different extraction methods (Twitter corpus).

In Figure 16, we illustrate the results of evaluating different models when they are trained on portions of the train set (10, 25 and 50 percent of DS), selected via random sampling without replacement, and tested on the whole test set. Similarly in Figure 17, we show the impact of training set size on different methods for Twitter corpus. Considering that our labeled DS corpus was relatively larger than labeled Twitter corpus, we could evaluate our extraction methods on smaller percentage of the DS training data (10, 25, 50 percent). However, for Twitter, we compared the performance of the system when trained on 50 percent of the labeled tweets, and compared the results with a system trained on the whole labeled tweets.

The general observed trend is that deep learning always remarkably increases the recall and the F-measure, with a cost of a relatively smaller decrease in precision. Interestingly, when less training data is available (e.g. 10% of DS train set in Figure 16, or 50% of

Twitter train set in Figure 17), we get a much larger improvement on recall and consequently the F-measure.

DeepHealthMiner Parameter Selection

The best results for the Twitter corpus achieved when we set the input window to include 7 tokens (including the target token). As illustrated in Figure 18, DeepHealthMiner achieved the highest F-measure on Twitter corpus when we set the number of hidden nodes to 200. We fixed the learning rate to 0.01 which is a value conventionally used in the similar NLP tasks in other domains. In Figure 20, we show the results of evaluation of hidden layer size for DS corpus. We varied the number of hidden nodes from 30 to 400 and found that 100 hidden nodes resulted in the best performance for DS.

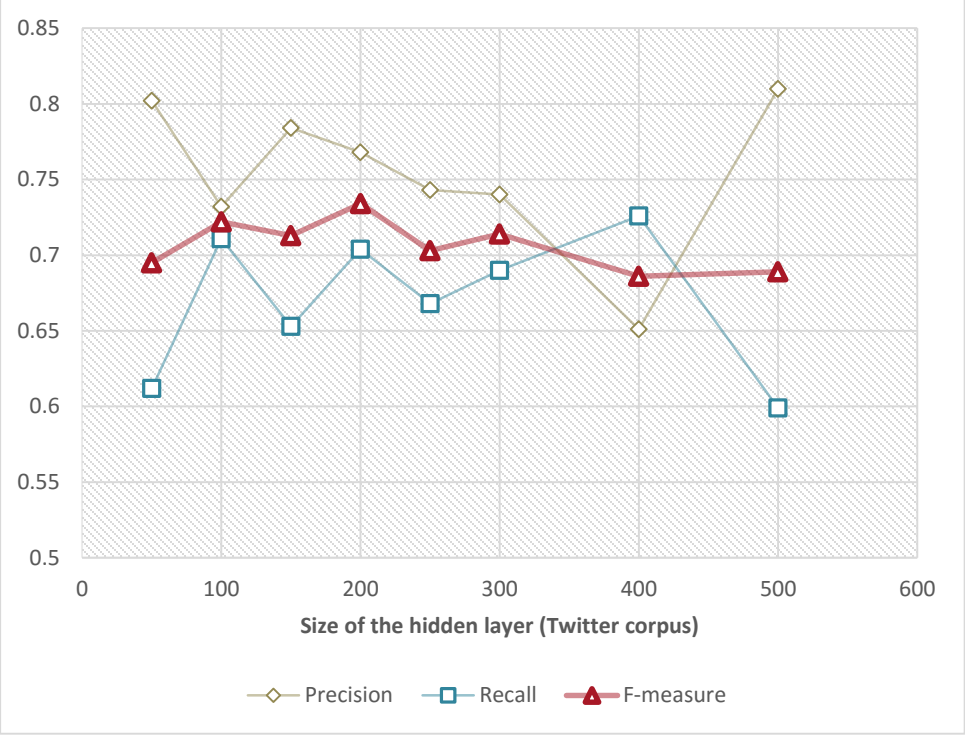


Figure 18: The impact of hidden layer size (# of nodes) on DeepHealthMiner extraction performance for Twitter corpus (context window size = 7, learning rate = 0.01).

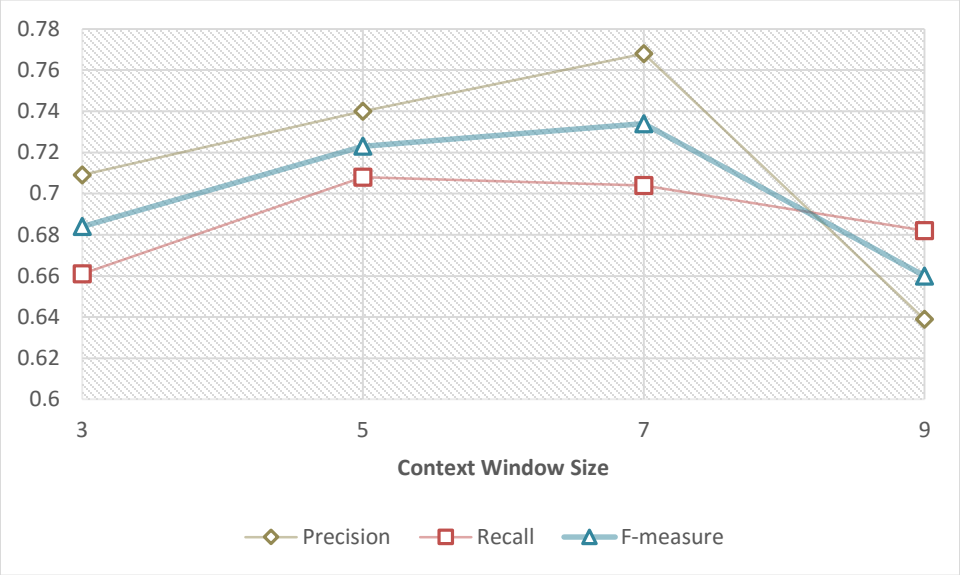


Figure 19: The impact of context window size on DeepHealthMiner extraction performance for Twitter corpus (size of the hidden nodes = 200, learning rate = 0.01).

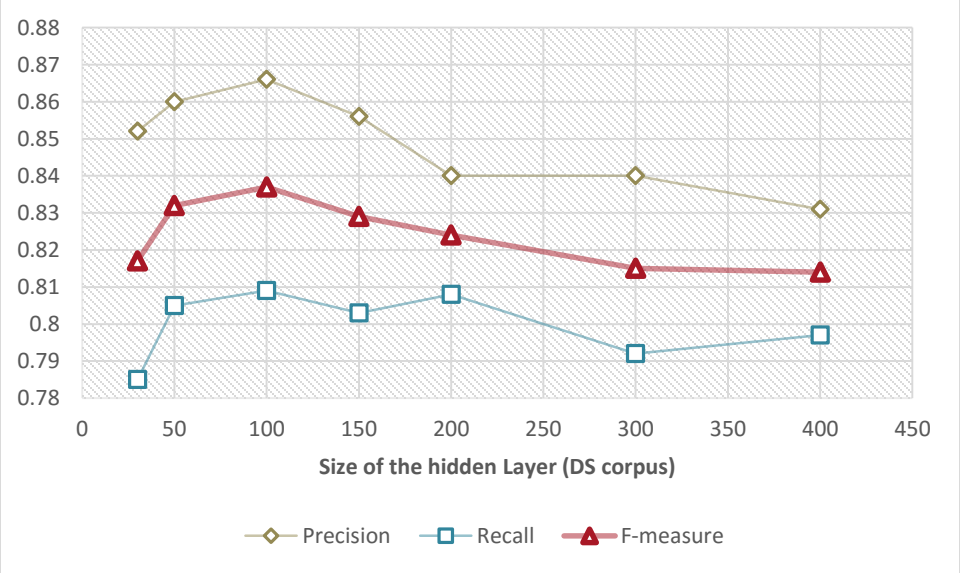


Figure 20: The impact of hidden layer size (# of nodes) on DeepHealthMiner extraction performance for DS corpus (context window size = 7, learning rate = 0.01).

Discussion

We found that the best performance in extracting health-related concepts from user posts in social media can be achieved by training a deep neural network classifier. DeepHealthMiner outperformed the strong baseline extraction techniques (Table 12).

We found that deep learning methods can remarkably reduce the need to use large volumes of labeled training data (Figure 16). When we trained the systems on training sets with different sizes, DeepHealthMiner performance was consistently higher than the baseline systems. This improvement was much noticeable for smaller training data (e.g. 10%). Interestingly when we only used 10% of the DS training set, DeepHealthMiner achieved a recall of 0.71 which is very close to the recall of the baseline CRF (0.72) when we used the whole train set (Figure 16 and Table 12).

By using deep learning compared to the conventional CRF, we achieved a larger F-measure improvement for the Twitter data compared with DS data. This can be justified by considering the type of sentences in Twitter. The twitter content, compared to DS, is shorter, less focused on health, and generally more informal. There are more irregularities in terms of sentence structures and expressed word. This makes the feature engineering more challenging for Twitter. It also affects the quality and correctness of the calculated features. On the other hand, deep learning automatically learns the classification features. Since a large volume of unlabeled user posts are available, the neural network automatically learns the best representation of user posts and achieves the highest recall and F-measure among other baseline methods.

Comparison of the Concept Extraction Methods

Lexicon-based Extraction

The MetaMap baselines performed poorly for this problem. Although MetaMap is the state-of-the-art concept extraction system from biomedical text [121], the results show the vulnerability of MetaMap when applied to informal text in social media. Evaluation of the baseline lexicon-based technique (Table 12) demonstrated that compared to MetaMap, it could extract ADR mentions with relatively high recall. The recall is anticipated to even further improve in future by augmenting the ADR lexicon with a larger subset of MedDRA entries and a more comprehensive list of common consumer expressions for ADRs. This relatively high recall indicates that the utilized lexicon-based techniques were effective in handling term variability in the user sentences. However, the method precision was relatively low which was mainly due to the matched mentions with entity types other than ADRs. When we used SVM to distinguish the entity types, the precision markedly increased, while the recall decreased but the overall extraction performance improved (Table 12). We utilized our proposed lexicon-based method for our preliminary experiment in normalizing the extracted medical concepts to the UMLS CUIs. One of the limitations of this normalization technique is that only mentions with overlapping tokens with the concept entries in the source lexicon can be normalized. Future research may evaluate the effectiveness of query expansion for normalization. The keyword tokens in the extracted mentions can be expanded by adding semantically similar tokens based on the context in the sentence before submitting the query to the indexed lexicon.

Patten-based Extraction

Using association pattern mining we could extract converging patterns in reporting ADRs in social media. The patterns were then used to extract ADR mentions from the test sentences. The performance of this technique is highly dependent to the richness of the training data, and the chance that the test set will include sentences that match the generated patterns.

There are several parameters that are important in the quality of the generated patterns. For instance, in generating the representative item sequences, one can use the token itself, the stemmed token or POS. We have the option of removing stop words or normalizing dates and numbers. Other parameters that can be set are the number of included previous and next words in the context and also the support and confidence values of the association rules. In this study we used DS train set for generating the patterns and applied them for both DS and Twitter. The performance of the patterns for tweets were relatively poor that can be explained by considering the noisy nature of the tweets. In our preliminary study [11] and showed that it is possible to partially address the limitations of the lexicon-based methods for extraction of ADRs from social media. Although we could extract complex ADRs using patterns, it required a lot of time and effort to improve the patterns' performance. However, these limitations could be addressed more efficiently, by using more advanced machine learning systems. Also, the current pattern mining approach automatically learns extraction patterns from the context and ignores the entity content itself. However, we could addressed this limitation by training more advanced systems such as ADRMine or DeepHealthMiner that are capable of training more complex models for concept extraction.

ADRMine and DeepHealthMiner

ADRMine and DeepHealthMiner both significantly outperformed the baseline CRF and other explored extraction methods. Since both proposed methods applied deep learning techniques and utilized the pre-trained word embeddings, the information from the large volume of unbaled data markedly improved the recall and F-measure. DeepHealthMiner improvement on the recall was consistently the highest among other methods. Although, ADRMine precision was generally higher than DeepHealthMiner particularly for smaller train set sizes, DeepHealthMiner F-measure was the highest in all experiments (Table 12, Table 13). Overall, we can conclude that DeepHealthMiner outperforms ADRMine particularly when the data is noisier and less labeled examples are available for training the system. Also, since DeepHealthMiner does not require feature calculation, the computational time is less than ADRMine.

Twitter vs. DailyStrength Corpus

As it is shown in Table 12, the extraction performance for DS is much higher than Twitter. This is partially related to the fact that there was less annotated data available for Twitter. In general, however, compared to DailyStrength extracting ADR information from Twitter poses a more challenging problem. Whereas DailyStrength is a health-focused site that fosters discussion from patients about their personal experiences with a drug, Twitter is a general networking site where users may be inclined to mention a particular drug and its side effects for any number of reasons. Some may include personal experiences, but others may tweet about side effects they heard about, be sharing of a news report, or a sarcastic remark. These nuances may be difficult for even annotators to

detect as the limited length of the tweets can make it more challenging for the annotator to ascertain the context of the mention. For instance, in this tweet: "*Hey not sleeping. #hotflashes #menopause #effexor*", it is difficult to determine whether the patient is taking the drug for their problem or if they are reporting ADRs.

6 CONCLUSION

User posts in social media are noisy and informal and medical concepts are often non-technical, descriptive, and complex to extract. This makes the concept extraction more challenging for social media compared to newswire, biomedical literature or clinical notes. This work proposed Natural language processing solutions for extraction of health-related concepts from user-generated content in social media. The proposed methods have been successful in addressing most of the challenges associated with medical NER from social media.

This work is the first attempt in applying deep learning for health information extraction from social media. We proposed ADRMine and DeepHealthMiner, two deep learning based methods for medical concept extraction. We also explored the effectiveness of several possible extraction techniques including lexicon-based, pattern-based and other machine learning based (support vector machine) methods.

The lexicon-based approach requires a list of medical concepts with associated UMLS concept IDs. Evaluation results for ADR extraction showed that this approach could achieve a relatively high recall but low precision. The low precision is primarily due to lack of a mechanism for distinguishing the entity types (e.g. ADRs vs indication) of the extracted mentions. To distinguish the types, we trained a multiclass SVM to learn to classify the entity type of the extracted mentions. This markedly improved the extraction performance.

The lexicon-based method has the advantage of providing immediate normalization of medical concepts since the extracted concepts were indexed along with UMLS CUIs. It also has the advantage of not requiring annotated training data. However, it has the

disadvantage of being limited to the concepts listed in the lexicon and complex consumer expressions remain undetected.

The pattern-based approach used association rule mining to identify the frequent patterns in expressing ADRs in user posts. It automatically learns extraction patterns from the labeled training sentences and then uses regular expressions to extract new mentions. This method has the advantage of not requiring a lexicon and enables extracting medical mentions that are not seen in the training data or listed in the lexicon. However, although the patterns extract mentions with high precision, they were limited to only very frequently observed items in the context, and resulted in a low recall.

ADRMine, the proposed machine learning based sequence tagger, achieved an F-measure of 0.82 for DailyStrength, and 0.72 for Twitter corpus, outperforming the baseline techniques. The effectiveness of various classification features explored in training the CRF model. We found that context and embedding clusters were the most contributing features. We utilized a large volume of unlabeled user posts for unsupervised learning of the embedding clusters, which enabled similarity modeling between the tokens, and gave a significant rise to the recall and the overall performance.

We explored the effectiveness of automatic feature learning for the task of health-related NER by introducing DeepHealthMiner, a deep learning pipeline for concept extraction. DeepHealthMiner employs a feedforward neural network sequence tagger for the NER task. For each token, the system receives a window of raw text including the target token and the context tokens, plus the corresponding embedding vectors. It then automatically learns more abstract features that are important for discriminating the possible tags. The results showed that the neural network classifier performed very well and remarkably

outperformed the baseline CRF classifier that uses standard NER features, and it even outperformed ADRMine, the state-of-the-art CRF sequence tagger that uses carefully engineered features including the embedding cluster features.

The deep learning sequence tagger that utilized large amounts of unlabeled data (around 3M sentences from user posts) and automatically learned word representation features was the most successful solution for the task of health-related concept extraction from social media. The postposed solutions are domain independent and potentially can be applied to other information extraction tasks if a large unlabeled data set, for training the word embeddings, and a relatively small labeled train set is available for the supervised learning.

The related resources including Twitter datasets and extraction software are made available at: http://diego.asu.edu/Publications/ADRMine.html.

Conclusions and Future Work

The proposed techniques and evaluations have shown that automatically generated word representations from unlabeled natural language text using deep learning have been very successful for the task of health information extraction from social media. When these word representations modeled as features in a state-of-the-art CRF classifier, the concept extraction performance significantly was improved compared to the baseline features. The system (ADRMine) outperformed several alternative extraction methods including a strong lexicon-based system. Interestingly, the performance even further improved, when a feedforward neural network that automatically learned classification features was employed for the task. We showed that both proposed deep learning solutions,

diminished the dependency on the size of the annotated data. The automatic feature learning of the neural network has been very successful in capturing discriminative features for extraction of new and creative consumer expressions from social media. Considering the rapidly increasing volume of user posts, and the fact that we generally have a comparatively small number of annotated sentences, deep learning based information extraction methods are anticipated to be very useful in future. Overall, the findings may largely facilitate the biomedical NLP research, given the difficulty of generating annotated corpora and considering that unlabeled data is often abundantly available. Moreover, we believe that the proposed extraction techniques are generalizable and can easily be applied to other entity types in health or other domains.

In this dissertation, we focused on concept extraction and only proposed preliminary solutions for the task of normalization that includes mapping an extracted mention to the corresponding concept in standard ontologies, such as UMLS and MedDRA. Normalization of medical concept in social media is relatively unexplored and future research should examine advanced machine learning and deep learning normalization techniques.

This work is the first step toward the next generation of deep learning NLP systems for analyzing health related information from social media. There are several potentially interesting studies on heath information extraction that can greatly benefit from the proposed concept extraction methods in this work. The future research may analyze user posts in social media to measure patient outcomes or drug effectiveness. In addition, information about drug off-label use and associated ADRs can be extracted from social media patient posts. Also future studies may investigate the medication effectiveness or

possible adverse reactions in patients who were excluded from drug clinical trials such as pregnant women, elderly or patients with comorbidities. Moreover, the extracted heath-related information in social media can be summarized to be used in automated question answering systems. Given that social media provides a different perspective over patient data that is different from clinical records, it is possible that unknown and new health-related information be extracted from it. This can then lead to new clinical hypothesis and studies that can validate the findings from social media.

REFERENCES

1 Tumasjan A, Sprenger T, Sandner P, *et al.* Predicting elections with Twitter: What 140 characters reveal about political sentiment. In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. 2010. 178–85. doi:10.1074/jbc.M501708200

2 Bollen J, Mao H, Zeng X. Twitter mood predicts the stock market. *J Comput Sci* 2011;**2**:1–8. doi:10.1016/j.jocs.2010.12.007

3 Pak A, Paroubek P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: *In Proceedings of the Seventh Conference on International Language Resources and Evaluation*. 2010. 1320–6. doi:10.1371/journal.pone.0026624

4 Ginsberg J, Mohebbi MH, Patel RS, *et al.* Detecting influenza epidemics using search engine query data. *Nature* 2009;**457**:1012–4. doi:10.1038/nature07634

5 Lampos V, Cristianini N. Tracking the flu pandemic by monitoring the social web. In: *2010 2nd International Workshop on Cognitive Information Processing*. Ieee 2010. 411–6. doi:10.1109/CIP.2010.5604088

6 Lamb A, Paul MJ, Dredze M. Separating Fact from Fear : Tracking Flu Infections on Twitter. In: *Proceedings of NAACL-HLT 2013*. 2013. 789–95.

7 Dredze M, Paul M, Bergsma S, *et al.* Carmen: A twitter geolocation system with applications to public health. In: *Expanding the Boundaries of Health Informatics Using Artificial Intelligence: Papers from the AAAI 2013 Workshop*. 2013. 20–4.

8 Merolli M, Gray K, Martin-Sanchez F. Health outcomes and related effects of using social media in chronic disease management: A literature review and analysis of affordances. *J Biomed Inform* 2013;**46**:957–69. doi:10.1016/j.jbi.2013.04.010

9 Paul MJ, Dredze M. You are what you Tweet: Analyzing Twitter for public health. In: *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*. 2011. 265–72. doi:10.1.1.224.9974

10 Leaman R, Wojtulewicz L, Sullivan R, *et al.* Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In: *Proceedings of the 2010 workshop on biomedical natural language processing*. 2010. 117–25.

11 Nikfarjam A, Gonzalez G. Pattern Mining for Extraction of mentions of Adverse Drug Reactions from User Comments. *AMIA Annu Symp Proc* 2011;**2011**:1019–26.

12 Banko M, Cafarella M, Soderland S, *et al.* Open information extraction for the web. In: *IJCAI*. 2007. 2670–6. doi:10.1145/1409360.1409378

13 Ritter A, Clark S, Etzioni O. Named entity recognition in tweets: an experimental study. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2011. 1524–34.

14 Lafferty J, McCallum A, Pereira F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the eighteenth international conference on machine learning, ICML*. 2001. 282–9.

15 Turian J, Ratinov L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010. 384–94.

16 Leaman R, Gonzalez G. BANNER: An executable survey of advances in biomedical named entity recognition. In: *Pacific Symposium on Biocomputing*. 2008. 652–63.

17 Liu X, Zhang S, Wei F, *et al.* Recognizing Named Entities in Tweets. In: *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2011. 359–67.

18 Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances In Neural Information Processing Systems*. 2012. 1097–105. doi:http://dx.doi.org/10.1016/j.protcy.2014.09.007

19 Deng L, Yu D. *Deep learning: methods and applications*. 2014.

20 Collobert R, Weston J, Bottou L, *et al.* Natural language processing (almost) from scratch. *J Mach Learn Res* 2011;**1**:2493–537.

21 Fox S, Duggan M. Health online 2013. 2013.

22 Ginsberg J, Mohebbi MH, Patel RS, *et al.* Detecting influenza epidemics using search engine query data. *Nature* 2009;**457**:1012–4. doi:10.1038/nature07634

23 Lee A. *Advese drug reactions*. 2001.

24 Pirmohamed M, James S, Meakin S, *et al.* Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. *BMJ* 2004;**329**:15–9. doi:10.1136/bmj.329.7456.15

25 Kongkaew C, Noyce PR, Ashcroft DM. Hospital admissions associated with adverse drug reactions: a systematic review of prospective observational studies. *Ann Pharmacother* 2008;**42**:1017–25. doi:10.1345/aph.1L037

26 Lazarou J, Pomeranz BH, Corey PN. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *Jama* 1998;**279**:1200.

27 van der Hooft CS, Sturkenboom MCJM, van Grootheest K, *et al.* Adverse drug reaction-related hospitalisations: a nationwide study in The Netherlands. *Drug Saf* 2006;**29**:161–8.

28 Leone R, Sottosanti L, Iorio ML, *et al.* Drug-related deaths: an analysis of the Italian spontaneous reporting database. *Drug Saf* 2008;**31**:703–13.

29 Anderson C, Krska J, Murphy E, *et al.* The importance of direct patient reporting of suspected adverse drug reactions: a patient perspective. *Br J Clin Pharmacol* 2011;**72**:806–22. doi:10.1111/j.1365-2125.2011.03990.x

30 Egberts TC, Smulders M, De Koning FH, *et al.* Can adverse drug reactions be detected earlier? A comparison of reports by patients and professionals. *BMJ* 1996;**313**:530–1.

31 van Grootheest K, de Graaf L, Berg LTW de J den. Consumer adverse drug reaction reporting: a new step in pharmacovigilance? *Drug Saf* 2003;**26**:211–7.

32 Aagaard L, Nielsen LH, Hansen EH. Consumer reporting of adverse drug reactions: a retrospective analysis of the Danish adverse drug reaction database from 2004 to 2006. *Drug Saf* 2009;**32**:1067–74. doi:10.2165/11316680-000000000-00000

33 Avery AJ, Anderson C, Bond CM, *et al.* Evaluation of patient reporting of adverse drugreactions to the UK 'Yellow Card Scheme':literature review, descriptive and qualitativeanalyses, and questionnaire surveys. 2011. doi:10.3310/hta15200. NIHR HTA, Southampton.

34 van Geffen ECG, van der Wal SW, van Hulten R, *et al.* Evaluation of patients' experiences with antidepressants reported by means of a medicine reporting system. *Eur J Clin Pharmacol* 2007;**63**:1193–9. doi:10.1007/s00228-007-0375-4

35 Vilhelmsson A, Svensson T, Meeuwisse A, *et al.* What can we learn from consumer reports on psychiatric adverse drug reactions with antidepressant medication? Experiences from reports to a consumer association. *BMC Clin Pharmacol* 2011;**11**:16. doi:10.1186/1472-6904-11-16

36 Hazell L, Shakir SAW. Under-Reporting of Adverse Drug Reactions. *Drug Saf* 2006;**29**:385–96. doi:10.2165/00002018-200629050-00003

37 Ginn R, Pimpalkhute P, Nikfarjam A, *et al.* Mining Twitter for Adverse Drug Reaction Mentions : A Corpus and Classification Benchmark. In: *proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (BioTxtM)*. Reykjavik, Iceland: 2014.

38 O'Connor K, Nikfarjam A, Ginn R, *et al.* Pharmacovigilance on Twitter? Mining Tweets for Adverse Drug Reactions. In: *American Medical Informatics Association (AMIA) Annual Symposium*. 2014.

39 DailyStrength. http://www.dailystrength.org/ (accessed 1 Jun2015).

40 Aramaki E, Miura Y, Tonoike M, *et al.* Extraction of adverse drug effects from clinical records. *Stud Heal Technol Inf* 2010;**160**:739–43. doi:10.3233/978-1-60750-588-4-739

41 Friedman C. Discovering Novel Adverse Drug Events Using Natural Language Processing and Mining of the Electronic Health Record. In: *AIME '09 Proceedings of the 12th Conference on Artificial Intelligence in Medicine: Artificial Intelligence in Medicine*. 2009.

42 Wang X, Hripcsak G, Markatou M, *et al.* Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *J Am Med Inform Assoc*;**16**:328–37. doi:10.1197/jamia.M3028

43 Gurulingappa H, Rajput A, Toldo L. Extraction of Adverse Drug Effects from Medical Case Reports. *J Biomed Semantics* 2012;**3**:1–4.

44 Toldo L, Bhattacharya S, Gurulingappa H. Automated identification of adverse events from case reports using machine learning. In: *Proceedings XXIV Conference of the European Federation for Medical Informatics. Workshop on Computational Methods in Pharmacovigilance*. 2012. 26–9.

45 Harpaz R, DuMouchel W, Shah NH, *et al.* Novel data-mining methodologies for adverse drug event discovery and analysis. *Clin Pharmacol Ther* 2012;**91**:1010–21. doi:10.1038/clpt.2012.50

46 Sarker A, Ginn R, Nikfarjam A, *et al.* Utilizing social media data for pharmacovigilance: A review. *J Biomed Inform* 2015;**54**:202–12. doi:10.1016/j.jbi.2015.02.004

47 Chee BW, Berlin R, Schatz B. Predicting adverse drug events from personal health messages. In: *AMIA Annual Symposium Proceedings*. 2011. 217–26.

48 Wicks P, Vaughan TE, Massagli MP, *et al.* Accelerated clinical discovery using self-reported patient data collected online and a patient-matching algorithm. *Nat Biotechnol* 2011;**29**:411–4. doi:10.1038/nbt.1837

49 Gurulingappa H, Rajput AM, Roberts A, *et al.* Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J Biomed Inform* 2012;**45**:885–92. doi:10.1016/j.jbi.2012.04.008

50 Yates A, Goharian N. ADRTrace: detecting expected and unexpected adverse drug reactions from user reviews on social media sites. *Adv Inf Retr* 2013;**7814 LNCS**:816–9.

51 Jiang L, Yang C, Li J. Discovering consumer health expressions from consumer-contributed content. In: *Social Computing, Behavioral-Cultural Modeling and Prediction*. 2013. 164–74.

52 Benton A, Ungar L, Hill S, *et al.* Identifying potential adverse effects using the web: A new approach to medical hypothesis generation. *J Biomed Inform* 2011;**44**:989–96. doi:10.1016/j.jbi.2011.07.005

53 Kuhn M, Campillos M, Letunic I, *et al.* A side effect resource to capture phenotypic effects of drugs. *Mol Syst Biol* 2010;**6**:343. doi:10.1038/msb.2009.98

54 Sampathkumar H, Luo B, Chen X. Mining Adverse Drug Side-Effects from Online Medical Forums. In: *2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology*. Ieee 2012. 150. doi:10.1109/HISB.2012.75

55 Yang CC, Yang H, Jiang L, *et al.* Social media mining for drug safety signal detection. In: *Proceedings of the 2012 international workshop on Smart health and wellbeing*. New York, USA: : ACM Press 2012. 33–40. doi:10.1145/2389707.2389714

56 Hakenberg J, Leser U. What makes a gene name? Named entity recognition in the biomedical literature. *Brief Bioinform* 2005;**6**:357–69.

57 Ingvaldsen JE, Özgöbek Ö, Gulla JA. Context-aware user-driven news recommendation. In: *INRA@ RecSys*. 2015. 33–6. doi:10.1017/CBO9781107415324.004

58 Nadeau D, Sekine S. A survey of named entity recognition and classification. *Lingvisticae Investig* 2007;**30**:3–26. doi:10.1075/li.30.1.03nad

59 Yann L, Yoshua B, Hinton G. Deep learning. *Nature* 2015;**521**:436–44. doi:10.1038/nature14539

60 Michelson M, Macskassy S. Discovering users' topics of interest on twitter: a first look. In: *Proceedings of the fourth workshop on Analytics for noisy unstructured text data. ACM*. 2010. 73–80.

61 Li C, Weng J, He Q, *et al.* TwiNER: named entity recognition in targeted twitter stream. In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM*. 2012. 721–30.

62 Wang K, Thrasher C, Viegas E, *et al.* An overview of Microsoft Web N-gram corpus and applications. In: *Proceedings of the NAACL HLT 2010*. 2010. 45–8.

63 Popescu A, Pennacchiotti M. Detecting controversial events from twitter. In: *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. New York, New York, USA: : ACM Press 2010. 1873. doi:10.1145/1871437.1871751

64 McCulloch W, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 1943;**5**:115–33.

65 Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature* 1986;**323**:533–6.

66 Jain A, Mao J, Mohiuddin K. Artificial neural networks: A tutorial. *IEEE Comput* 1996;**29**:31–44.

67 Amari S. Natural gradient works efficiently in learning. *Neural Comput* 1998;**10**:251–76.

68 Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell* 2013;**35**:1798–828. doi:10.1109/TPAMI.2013.50

69 Elman J. Finding Structure in Time. *Cogn Sci* 1990;**211**:179–211.

70 Bengio Y, Ho Jean-Sébastien Senécal, Fréderic Morin and J-LG. A Neural Probabilistic Language Model. *J Mach Learn Res* 2003;**3**:1137–55.

71 Goodman JT. A bit of progress in language modeling. *Comput Speech Lang* 2001;**15**:403–34. doi:10.1006/csla.2001.0174

72 Schwenk H, Gauvain J. Connectionist language modeling for large vocabulary continuous speech recognition. In: *{Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. 2002. 765–8.

73 Mikolov T. *Statistical Language Models Based on Neural Networks*. 2012.

74 Mikolov T, Kopecky J. Neural network based language models for highly inflective languages. In: *IEEE International Conference on Speech and Signal Processing*. 2009. 2–5.

75 Mikolov T, Chen K, Corrado G, *et al.* Efficient estimation of word representations in vector space. *arXiv Prepr arXiv13013781* 2013.

76 Hofmann T. Probabilistic latent semantic analysis. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999. 50–7.

77 Blei D, Ng A, Jordan M. Latent dirichlet allocation. *J Mach Learn Res* 2003;**3**:993–1022.

78 Mikolov T, Karafiát M, Burget L, *et al.* Recurrent neural network based language model. In: *INTERSPEECH*. 2010. 1045–8.

79 Le Q, Mikolov T, Com TG. Distributed Representations of Sentences and Documents. 2014;**32**.

80 Bengio Y. Learning deep architectures for AI. *Found Trends Mach Learn* 2009;**2**:1–127.

81 Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th international conference on Machine learning. ACM*. 2008. 160–7.

82 Mikolov T, Yih W, Zweig G. Linguistic regularities in continuous space word representations. *Proc NAACL-HLT* 2013.

83 Mundra R, Socher R. Deep Learning for NLP Lecture Notes Part III. 2016.http://cs224d.stanford.edu/lecture_notes/notes3.pdf

84 ADRMine Resources. DIEGO Lab, Dep. Biomed. Informatics, Arizona State Univeristy. 2015.http://diego.asu.edu/Publications/ADRMine.html

85 Pimpalkhute P, Patki A, Nikfarjam A, *et al.* Phonetic spelling filter for keyword selection in drug mention mining from social media. In: *AMIA Joint Summits on Translational Science proceedings AMIA Summit on Translational Science*. 2014. 90–5.

86 MongoDB. 2014.https://www.mongodb.com/

87 Sarker A, O'Connor K, Ginn R, *et al.* Pharmacovigilance from social media: Annotation guidelines. 2015.

88 Cohen J. A Coefficient of Agreement for Nominal Scales. *Educ Psychol Meas* 1960;**20**:37–46.

89 Viera AJ, Garrett JM. Understanding Interobserver Agreement : The Kappa Statistic. *Fam Med* 2005;**37**:360–3.

90 Twitter Adverse Drug Reaction Corpus. DIEGO Lab, Dep. Biomed. Informatics, Arizona State Univeristy. 2015.http://diego.asu.edu/downloads/publications/ADRMine/download_tweets.zip

91 SIDER 2 — Side Effect Resource. http://sideeffects.embl.de/

92 Zeng-Treitler Q, Goryachev S, Tse T, *et al.* Estimating consumer familiarity with health terminology: a context-based approach. *J Am Med Informatics Assoc* 2008;**15**:349–56. doi:10.1197/jamia.M2592.Introduction

93 ADR Lexicon. DIEGO Lab, Dep. Biomed. Informatics, Arizona State Univeristy. 2015.http://diego.asu.edu/downloads/publications/ADRMine/ADR_lexicon.tsv

94 Mikolov T. Word2vec. 2013.https://code.google.com/p/word2vec/

95 Nikfarjam A, Sarker A, O'Connor K, *et al.* Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *J Am Med Informatics* 2015.

96 Okazaki N. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). 2007.http://www.chokkan.org/software/crfsuite/

97 Apache Lucene. http://lucene.apache.org/

98 Atkinson K. SCOWL (Spell Checker Oriented Word Lists). http://wordlist.aspell.net/. 2014.

99 Zhou X, Zhang X, Hu X. Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining. In: *proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. 2007. 197–201.

100 Miller G a. WordNet: a lexical database for English. *Commun ACM* 1995;**38**:39–41. doi:10.1145/219717.219748

101 Manning CD, Klein D. Accurate Unlexicalized Parsing. In: *Proceedings of the 41st Meeting of the Association for Computational Linguistics*. 2003. 423–30.

102 Kilicoglu H, Bergler S. Syntactic dependency based heuristics for biological event extraction. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. 2009. 119–27.

103 Nikfarjam A, Emadzadeh E, Gonzalez G. A Hybrid System for Emotion Extraction from Suicide Notes. *Biomed Inform Insights* 2012;**5**:163. doi:10.4137/BII.S8981

104 Taboada M, Brooke J, M T, *et al.* Lexicon-based methods for sentiment analysis. *Comput Linguist* 2011;**37**:267–307.

105 Socher R, Perelygin A, Wu J, *et al.* Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In: *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. 2013.

106 Penn Treebank II Constituent Tags. http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html (accessed 1 Jan2016).

107 Attardi G. Deepnl: a deep learning nlp pipeline. In: *Proceedings of NAACL-HLT*. 2015. 109–15.

108 Lopez V, Sabou M, Motta E. PowerMap: mapping the real semantic web on the fly. In: *International Semantic Web Conference*. 2006. 414–27.

109 Emadzadeh E, Nikfarjam A, Ginn R, *et al.* Unsupervised Gene Function Extraction using Semantic Vectors. *Database* 2014.

110 Huang M, Liu J, Zhu X. GeneTUKit: a software for document-level gene normalization. *Bioinformatics* 2011;**27**:1032–3. doi:10.1093/bioinformatics/btr042

111 Joachims T. Text categorization with Support Vector Machines: Learning with many relevant features. In: *European conference on machine learning*. Berlin/Heidelberg: : Springer-Verlag 1998. 137–42. doi:10.1007/BFb0026683

112 Joachims T. Making large scale SVM learning practical. In: *Advances in Kernel Methods - Support Vector Learning,B. Schölkopf and C. Burges and A. Smola (ed.)*. MIT-Press 1999.

113 Agrawal R, Srikant R. Fast algorithms for mining association rules. In: *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*. Citeseer 1994. 487–99.

114 Nakhaeizadeh G, Hipp J, Güntzer U. Algorithms for association rule mining --- a general survey and comparison. *ACM SIGKDD Explor Newsl* 2000;**2**:58–64. doi:10.1145/360402.360421

115 Borgelt C. Apriori - Association Rule Induction / Frequent Item Set Mining. 2010.http://www.borgelt.net/apriori.html (accessed 1 Jan2011).

116 Tsai RT-H, Wu S-H, Chou W-C, *et al.* Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics* 2006;**7**:92. doi:10.1186/1471-2105-7-92

117 Sarker A, Nikfarjam A, Gonzalez G. PSB 2016 Social Media Mining Shared Task Workshop. 2016. http://diego.asu.edu/psb2016/task2data.html

118 Yeh A. More accurate tests for the statistical significance of result differences. In: *Proceedings of the 18th conference on Computational linguistics*. 2000. 947–53.

119 Pado S. User's guide to sigf: Significance testing by approximate randomisation. 2006.

120 Jonnalagadda S, Gonzalez G. Sentence Simplification Aids Protein-Protein Interaction Extraction. *arXiv Prepr arXiv10014273* 2010;:6.

121 Leaman R, Dogan RI, Lu Z. DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics* 2013;**29**:2909–17. doi:10.1093/bioinformatics/btt474