DRUG ADVERSE EVENTS CLASSIFICATION USING SOCIAL MEDIA CONTENT

Ranith Sachintha Ranawaka

(179346C)

Degree of Master Of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2021

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic, or another medium. I retain the right to use this content in whole or part in future works.

..... **.....**

Ranith Sachintha Ranawaka Date

The above candidate has carried out research for the Master's thesis/ Dissertation under my supervision.

..... **.....**

Dr. Surangika Ranathunga Date

Acknowledgements

I would like to express profound gratitude to my advisor, Dr. Surangika Ranathunga, for her invaluable support by providing relevant knowledge, materials, advice, supervision, and useful suggestions throughout this research work. Her expertise and continuous guidance enabled me to complete my work successfully. Further, I would like to thank Mr. Sanmugan Aravinthan, for providing valuable resources for this project.

I am grateful for the support and advice given by Dr. Uthayasanker Thayasivam. Further, I would like to thank all my colleagues for their help in finding relevant research material, sharing knowledge and experience, and for their encouragement.

I am as ever, especially indebted to my parents and family for their love and support throughout my life. Finally, I wish to express my gratitude to all my colleagues at IQVIA Sri Lanka, for the support given to me to manage my MSc research work.

Abstract

On-time detection of possible adverse events a drug may have has been a critical issue for the pharmaceutical industry, although it undergoes rigorous clinical trials there still can be adverse effects once it reaches the market, this is known as post-market drug safety surveillance. The ordinary way to collect these was through physicians who prescribe the drug reporting back to the pharmaceutical company. But this process consumes time and has the risk of missing important drug adverse reactions.

The recent popularity of social media has led people to communicate extensively about their aspects in day-to-day life, this includes the communications of the experience regarding the drugs and their adverse events. This makes social media a rich resource for monitoring drugs after they reach the market.

In this research, we experiment with machine learning models including deep learning models using social media contents manually verified by health care professionals for the presence of drug adverse events. The Social media data has been acquired through popular health care social media channels from their respective APIs.

Well-known Text classification algorithms such as SVM and Logistic Regression provide the best accuracy for ADR mining, CNN's which has recently shown high accuracy levels for text classification also shows high levels of accuracy for ADR classification tasks.

Table of Contents

List of Figures **Page**

List of Tables

List of Equations

List of Abbreviations

Abbreviation	Description
ADR	Adverse Drug Reaction
OTC	Over the Counter
FDA	Food and Drug Administration
DoTs	Dose Time and Susceptibility
PV, PHV	Pharmacovigilance
CRM	Customer Relationship Management
URL	Universal Resource Locator
POS	Part of Speech
UMLS	Unified Medical Language System
SVM	Support Vector Machines
ME	Maximum Entropy
MNB	Multinomial Naïve Bayes
CTakes	Clinical Text Analysis and Knowledge Extraction System
jSRE	Java simple relation extraction
NLP	Natural Language Processing
BOW	Bag of Words
TF	Term Frequency
BOAW	Bag of Audio Words
TF-IDF	Term Frequency Inverse Document Frequency
SNOMED CT	Systematized Nomenclature of Medicine - Clinical Terms
ICH	International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human
CRF	Conditional Random Fields
RNN	Recurrent Neural Network

CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory

CHAPTER 1: INTRODUCTION

1.1 **Drugs and Clinical Trials**

A drug or medication is used to either diagnose, cure, prevent or treat a disease, it is one of the most important parts of the field of medicine, and the science behind developing these drugs is known as pharmacology. Drugs can be mainly classified into two, namely:

- Prescription Drugs – A drug dispensed by a pharmacy only with a prescription from a qualified physician or a nurse.
- OTC (Over the counter drugs) – Drugs that can be obtained from groceries or pharmacies without any prescription from a doctor are usually very basic drugs such as Aspirin.

Producing a successful drug usually takes years of research with top-class researchers, and there is a huge cost associated with it. After producing a drug there are a vast number of tests that it should go through as well as it should comply with many rules and regulations enforced by the government and relevant authorities.

There are two main methods of testing a drug, namely clinical studies, and clinical trials [1]. Clinical studies also known as observational studies observe people in normal settings after grouping people into different categories by researchers and then compare changes over time. Clinical trials are aimed at evaluating medical, surgical, or behavioral intervention of people taking part, they often check on new drugs or devices and assess whether the new treatment has fewer side effects and more effective compared to the standard treatment.

Before a drug starts to be tested on humans, the FDA (Food and Drug Administration), which is the government body governing drug manufacturing in the US should approve the tests. The FDA does this approval based on laboratory studies done on animals to test a therapy's safety and efficacy.

1.2 Drug adverse reactions

Detecting adverse reactions that occur from drugs has become a very important task in today's pharmaceutical and healthcare industry. Rules and regulations are being brought up governing the field of adverse events that occur from drugs, specifically by the FDA. In the United States, Pharmaceutical companies are responsible for documenting and reporting adverse events related to the drugs they manufacture.

A drug adverse reaction is an unintended and harmful event that occurs due to the usage of the drug. Any drug will have the potential of causing an ADR (Adverse Drug Reaction) [2]. In the United States, around 5% of all hospital admissions are the result of an ADR, and around 10%–20% of inpatients will have at least one ADR during their hospital stay [2], but the exact amount of deaths due to ADRs cannot be confirmed because these patients have had other forms of severe health conditions.

ADR's can be divided into two categories as follows:

- Adverse Events – A negative experience encountered by an individual, may or may not be due to the drug.
- Serious Adverse Events – Any fatal adverse event, life-threatening, or results in hospitalization.

Certain drugs have been reported as causing more ADR than other drugs, they include antiplatelets, anticoagulants, cytotoxic, immunosuppressant, diuretics, antidiabetics, and antibiotics [1].

Depending on the underlying cause drug adverse reactions can be divided into two categories [2]:

- Type A: Are dependent on the dose and are predictable based on the pharmacology of the drug.
- Type B: Are completely random reactions that are not based on the pharmacology of the drug.

But this basic classification does not work for ADRs, there are other cases where ADR happens like adverse effects associated with cumulative drug exposure or withdrawal reactions. A more comprehensive classification scheme is DoTs (Dose Time and Susceptibility), which classifies reactions dependent on the Dose of the drug, the time course of the reaction, and relevant

susceptibility factors (such as genetic, pathological, and other biological differences [3]). As well as classifying reactions, DoTS has the advantage of being helpful to consider the diagnosis and prevention of ADRs in practice.

The standard way drug ADRs are reported is through physicians, the prescribing physician comes to know that a patient has some adverse effects due to a drug he has prescribed, and then he is responsible for reporting that as an adverse event if there is enough information that the effect the patient has is related to the drug prescribed.

1.2 Pharmacovigilance

Pharmacovigilance commonly known as drug safety, PV, or PHV is the collection, detection, assessment, monitoring, and prevention of adverse effects of pharmacological drugs [4]. The disaster caused by the drug thalidomide in 1961 [5] which made infants congenitally deformed due to the exposure of unsafe medicine taken by pregnant mothers caused the international community to make systems that address drug safety issues systematically.

The international communities took steps to make standardized adverse event reporting systems so that every incident occurring will be captured and made available under a centralized location.

Although a drug goes under heavy clinical trials that mimic real-world drug usage, there are considerable differences between the two scenarios that bring the need for pharmacovigilance practices.

There have been instances where big players in the pharmaceutical industry have had to withdraw drugs from the market due to the reporting of serious adverse events [6].

A few of the aims in bringing pharmacovigilance practices are summarized as follows:

- To improve the safety and patient care about the use of medicine and all other medical and paramedical activities.
- Improving public safety regarding the use of drugs.
- Assessing the benefits, harm, effectiveness, and risks so that more safety can be ensured, and they can be made more effective.

- Promoting education and awareness about pharmacovigilance to the public and health care professionals.
- Insufficient evidence of safety from clinical trials needs for post-market surveillance.
- Detects problems related to medicines, allows for communication of problems on time.
- Limitations of data availability in clinical trials.

1.4 Problem/Opportunity

Although the physicians are supposed to report possible drug ADRs it is rare that a patient comes back and reports it to the physician, hence most of the drug ADRs go unreported which can be very dangerous. Finding these unreported adverse events is a major challenge for pharmaceutical companies, but lately, people tend to post these extensively on social media [7]. Thus, pharmaceutical companies must analyze through unstructured data like Facebook comments, tweets, call center records of health insurance companies, and CRM (Customer Relationship Management) systems of pharmaceutical company sales representatives, etc.

The data that is available in social media is highly important as it provides an interaction with the patient on a personal level, but this data is massive and manual evaluation to track down ADRs is a very labor-intensive task. Machine-learning based approaches can be used to analyze the data and flag the data that may potentially contain adverse events which can be further analyzed by professionals to confirm, this way it will reduce the human error that happens when going through that amount of data and it will also reduce the amount of expert labor required. The FDA states their regulation as an ADR should be reported within 24 hours of occurring [8], but with these manual methods achieving this timeline is impossible.

As a solution companies like IQVIA [9] has come up with products like AE Tracker[10] which scans social media and forums and blogs, but they use a medical ontology to detect adverse events from social media, this approach has a very high false-positive rate as well as the ontology should be kept up to date.

There is a considerable research effort in classifying whether social media content is an adverse event or not but the automatic extraction of this adverse effect from the social media content is not

much done. Also, mostly suboptimal ways such as Support Vector Machines and Naive Bayes have been commonly used to automatically learn classifying drug ADR related data from social media data. There have been very few efforts on employing modern methods such as Deep Learning and Word embeddings for mining drug ADRs from social media data [11,12,13,14].

This research has considered only popular social media platforms such as Twitter and Facebook, but more valuable information exists in other social media platforms such as Reddit [9], also the existing research has only experimented on a limited dataset such as a predefined set of drugs or around a certain set of diseases not generally from health-related data on Social media.

1.5 Objectives

The overall objective of this research is to accurately mine drug adverse events from Social media data, it can be divided into the following sub-objectives:

- Creating a representative dataset to train and test a machine learning model from health care data across all popular social media platforms.
- Experiment with text classification and deep learning methodologies to accurately classify ADR's

1.6 Contributions

We have experimented with the biggest generic health dataset obtained from Social media containing 45000 odd tweets and Facebook posts/comments, these were also manually annotated by health care professionals.

We were able to obtain the best classification results so far in mining drug adverse events from Social media content. We also experimented in-depth with different parameters of SVM & Logistic Regression where similar research in this area did not try experimenting with.

1.7 Thesis Structure

Chapter 1 of this Thesis introduces drug adverse reactions and the problem that we are addressing in this research. A literature review on this research area has been done in Chapter 2 highlighting key findings in each of the research we investigate. Chapter 3 details the methodology we followed in this research along with the high-level architecture. The system evaluation is presented in Chapter 4 along with the results of each experiment we conducted. We conclude this Thesis with our conclusions, summary, and possible future efforts in Chapter 5.

CHAPTER 2: LITERATURE REVIEW

With the increasing popularity of social media, people have started posting their healthcare issues on social media platforms, this data is electronically available. Researching on ADR detection from this digital data has gained attention. The recent developments of advanced machine learning, and natural language processing techniques also is a reason for research interests in this field.
The following sections summarize different aspects of using Machine Learning and Natural Language Processing on the detection of ADRs from Social Media and details about past research on these areas.

2.1 Definition

Table 2.1 shows three instances of health-related social media posts, the first is a clear ADR mention as it mentions due to a chemical in the medication there is a side effect. The second and the third seems as an ADR but not in reality as it mentions only about a medication not answering properly and hence not an ADR.

Class	**Content**
ADE	It has pseudoephedrine in it. It does that. Ups hr and bp
Non-ADE	Waste of money to me! It did"t work I have this in many forms and none of them work
Non-ADE	My face is on fire and Tylenol isn't helping. I'm burning up. Fingers crossed I'm not getting sick.

Table 2.1.1: ADE and Non-ADE Examples

When classifying ADR, we should be precautious in distinguishing an adverse event and mention the actual event in which a drug was prescribed, following is an example:

Class	**Content**
ADR	Well if you don't want your kids to choke to death in the middle of the night on their phlegm, then do not give them Mucinex for kids. It's a good thing I woke up to check on my daughter when I did because who knows what could have happened! Mucinex produces more phlegm than you even have in the first place.
Non - ADR	Mucinex works, but it takes time. It makes it easier to cough. I sounded like I had a three-pack a day habit without it.

Table 2.1.2: Distinguishing ADR and Non-ADR

In table 2.1.2, the first is an ADR since the drug seems to worsen the condition as per the statement. The second one mentions an adverse condition too, but the drug has been given to treat the condition. We discuss more on extracting the relationship between the drug and the adverse effect later in this chapter.

2.2 Data Preprocessing Techniques

Social media data contains a lot of slang, symbols emojis, etc. These need to be filtered out before extracting features to determine ADR. Apart from this, there are general preprocessing techniques that can be used when text mining. Stop words are usually removed hence they do not provide meaning when taken individually, but past research on ADR detection from Social media data suggests that when taking word 2-grams and 3-grams as features stop words has a positive effect [15].

POS (Part of Speech) tagging is also done as a pre-processing step, here POS tags are assigned to each word and these tags are then used for classification since the language used in Twitter is different from the usual written language, POS tagging specific to online conversations [16] has been used in past research. We can also use general POS taggers such as Stanford POS tagger [17], it is a lexicalized probabilistic parser that provides various information such as the syntactic structure of text segments, dependencies, and POS tags.

Other preprocessing techniques involve tagging text using external vocabularies such as UMLS (Unified Medical Language System) [18], removing social media-specific entities such as hashtags and special characters such as @ symbols, removing possible advertisements since data from Social media contains a lot of advertisements and marketing material and stemming and lemmatization to get the base forms of words [19].

2.3 Binary Classification of ADR and Non-ADR Text

Social media generates data at a very fast rate, identifying whether a piece of text contains an ADR or not is the first step in mining ADRs from social media. The popular research approach has been to use supervised classifiers that use different features related to the task at hand. Research done by Abeed et al. [15] uses Naïve Bayes, SVM(Support Vector Machines), and ME(Maximum Entropy) classifiers to classify tweets into ADR and non-ADR.

SVM has been the natural choice for text classification because of its ability to deal with high-dimensional feature space. Ofogh et al. [19] have used an SVM and an MNB (Multinomial Naïve Bayes) classifier for binary classification of comments from DailyStrength which is a healthcare-related forum. Sarkar et al. [20] have used SVM, MNB, ME, and a tree-based classifier known as J48 to classify tweets containing possible mentions of drug abuse and then used stacking [21], a technique where the predictions from the different classifiers are combined and another algorithm is trained to make a final decision based on the individual predictions.

Most of the past research for binary text classification has either been using SVM or the ME classifier or an ensemble of both using a weighted average [19,22,23].

The corpora-based binary classification has also been researched, but the issue here is that this corpus would suit if we are considering the adverse reactions of a specific drug or set of drugs, but not if we are trying to build a generic model for mining ADRs from Social Media streams. Karen

et al. [24] have used a corpus of tweets for 74 drugs, the first step was a binary annotation of each tweet indicating whether it contained an ADR mention or not. A combination of the tweets identified as containing an ADR and a randomly selected sample of 1,000 tweets that were marked as not containing a mention of an ADR were then annotated for specific concept spans and UMLS [18] concept IDs.

UMLS is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems. A key problem when mining ADRs from text is that distinguishing mentions about adverse reactions (an unexpected, negative effect resulting from the adequate use of the drug) from mentions about indications (the sign or symptom for which the drug was prescribed in the first place).

2.4 Drug and ADR Relationship Extraction

Drugs and possible adverse reactions can be identified separately from the text, but the adverse reaction should occur due to the drug which is then called a drug adverse reaction. Many approaches have been used to extract ADR relations with drugs from unstructured text. The most common approach is based on the co-occurrence of entities it assumes that if two entities occur in the same sentence there is a high chance that these two are related, this approach has high recall but low precision [25]. Since these do not consider linguistic characteristics this method is very fast and often taken as a baseline to evaluate performance [26,27].

Rule-based methods are also used to extract relationships between ADRs and drugs, the rules are defined manually using features from the context in which the relations of interest occur. Such features may be prefixes and suffixes of words, POS tags, chunking information, etc. [28,29,30]. This can increase precision but achieves a lower recall.

Machine learning approaches automatically build classifiers using contextual features that they get using Natural Language Processing (NLP) techniques, such as shallow parsing, which divides the sentence into chunks [31,32], or full dependency parsing, which provides a complete syntactic analysis of sentence structures [33]. But these require annotated training sets and the processing time may be very high.

Apache CTakes (clinical Text Analysis and Knowledge Extraction System) [34] is an open-source NLP system that extracts clinical information from electronic medical records. It can identify medical named entities such as Drug names, diseases or disorders, symptoms, whether words are negated or not, etc. Tafti et al. [11] have used a binary classifier to classify sentences into ADR and non-ADR categories and then used CTakes to determine drug and ADR relationships in the sentences categorized as containing ADRs.

jSRE (Java simple relation extraction) is an open-source relation extraction system, it uses a supervised machine learning method. jSRE uses a combination of kernel functions to integrate two different information sources, namely the whole sentence where the relation appears and the local contexts around the interacting entities. Harsha et al. [35] have used jSRE to extract drug and event relationships in sentences classified as containing ADRs.

The feature-based method creates a feature vector in an n-dimensional space, features should be selected to define the characteristics of data, Yang et al. [36] used bag-of-words features, medical-related words, and entity distance as features. Bui et al. [37] adopted bag-of-words features, part-of-speech tag features, and entity distance features to classify relation types.

Kernel-based methods are an effective alternative to explicit feature extraction, it retains the original representation of relation instances and uses the object only via computing a kernel function between a pair of instances. Tree kernels leverage the dependency tree of the entire sentences [38], while the shortest dependency path kernels only concern with the minimal proportion that bridges two entities [39]. They assume that the most important features for relation extraction concentrated on the shortest path between two entities on the dependency tree.

Already built tools are also available for concept extraction from unstructured texts, Peregrine is such a system, Peregrine is a dictionary-based concept recognition and normalization tool developed at the Erasmus University Medical Center [40]. It finds concepts by dictionary lookup, performs word-sense disambiguation if necessary, and assigns CUIs (concept unique identifiers). Ning et al. [41] have used the Peregrine system along with an NLP (Natural Language Processing)

system to further improve the classification. The NLP module had five basic sub-modules as follows:

- The coordination module uses POS and chunking information to reformat the coordination phrase and feed the formatted text into the concept normalization system for proper annotation of the concepts.
- The abbreviation module combines an abbreviation expansion algorithm with POS and chunking information to improve the recognition of abbreviations.
- The term variation module contains several rules that adjust noun phrases and feed the adjusted phrase into the concept normalization system again, to check whether it refers to a concept.
- The boundary correction module contains several rules that correct the start and end positions of concepts identified by the system, based on POS and chunking information.
- The concept filtering module consists of two rules that suppress concepts that were identified by the concept normalization system.

Hybrid approaches that combine machine learning and rule-based methods have also become popular in recent years [42,43].

2.5 Feature Engineering

Selecting the right set of features is crucial in the accuracy achieved using machine learning, since ADR detection is done from the text the commonly used features for text extraction have been used for this too.

Huh et al. [44] considered three feature types word unigram, sentiment analysis features, and the thread length. Karen et al. [24] too has used a CRF (Conditional Random Fields) algorithm, in addition to the above-mentioned features they have used word negations too. Abeed et al. [15] use word-N-grams, UMLS semantic types, syn-set expansion, change phrases, matches on an ADR Lexicon, Sentiwordnet scores, and also topic modeling as features. Synset (Synonym Set) expansion has been used because past research has shown that certain terms play important roles in determining the polarity of the whole sentence. For each adjective, noun, or verb in a sentence,

WordNet [45] is used to identify the synonyms of that term and add the synonymous terms, attached with the SYN tag, as features, then the TF-IDF measure is obtained for each derived synonym. Change phrases [46] determine whether a sentence represents positive information or negative information based on the change that happens. If a bad thing was reduced, then it is a positive outcome, if a bad thing was increased, then the outcome is negative. This feature set attempts to capture cases when a good/bad thing is increased/decreased. Topic modeling attempts to discover abstract topics that occur in collections of texts. The intuition is that ADR assertive text segments are likely to exhibit specific abstract topics. Drug dosage information has been used too as a feature in some research, this is because to determine that an event was caused by a drug there should be a specific dosage and the frequency, ex. two tablets per day.

Below are few commonly used feature extraction techniques used in general text processing as well as ADR extraction.

2.5.1 Word Embeddings

Word Embeddings is the representation of a documented vocabulary in the form of a vector, it can capture the semantic and syntactic similarity between neighboring words, the context of a word in a document, etc. Numerous researches have shown that using word embeddings boosts the accuracy of NLP tasks such as syntactic parsing and sentiment analysis. Conceptually word embeddings involve a mathematical embedding that starts from one dimension per word to a continuous vector space with lower dimensions. There are several methods that we can use to generate this mapping including neural networks, dimensionality reduction on the word co-occurrence matrices, and probabilistic models. Phrase embeddings are a recent development that has been used in research like word embeddings but using phrases.

Xiao et al. [13] have studied the effect of the background dataset used to train the embedding models, the context window size, and the dimensionality of word embeddings on the classification performance, they use a corpus from Wikipedia and one from Twitter to classify election-related tweets and show that the corpora trained with Twitter perform better. They also claim that using

larger context windows and dimension sizes in word embeddings improves the performance when used with CNN.

Word2Vec is a popular implementation of word embeddings by Google [47]. In Word2Vec, we are trying to predict a given word based on its context, or predicting a surrounding context based on a given word. Kathy et al. [12] have used supervised phrase embeddings to learn region embeddings of high-level task-relevant concepts, the learning goal has been to preserve the predictive structure of these text regions and learn their low dimensional vector representations, many of the health conditions that describe an adverse drug event are multi-word phrases and learning their vector representations can be beneficial for ADR classification.

Ahmad et al. [11] have used Biomedical articles and Health-related social media blog posts and built a word2vec representation of every word/sentence and then grouped the similar vectors training words/sentences against other words/sentences that neighbor them in the input text corpus. When a new text comes it calculates the cosine distance similarity between the new sentence and the training vector and decides the best fitting class for the sentence.

2.5.2 BOW (Bag of Words)

The most common feature extraction for NLP tasks is the BOW approach. In bag-of-words, we look at the histogram of word occurrences in a given corpus, we disregard the grammar and word order but keep the frequency of the word occurring. From a BOW model mainly, we can retrieve the term frequency (TF) of each word. The main drawback of BOW representation is that it's discrete and cannot capture the semantic relationship between words, hence it is hard to find BOW being used for modern-day text classification research, but it has variations like Bag of Audio Words (BOAW) which has shown promising results in areas like audio classification [48].

2.5.3 TF – IDF (Term Frequency Inverse Document Frequency)

TF-IDF (Term Frequency Inverse Document Frequency) value shows how important a word is to the document in a collection of documents. TF-IDF value increases proportionally to the number of times a word appears in a document and is offset by the number of documents the word contains, it captures the fact that frequently occurring words have more relevance in a document and also negates out commonly occurring words in general, it is one of the popular term weighting schemes today, most of the recommender systems use it [49], Abeed et al. and Huh et al.[15, 44] use TF-IDF on health-related texts. However, TF-IDF is also unable to capture the semantic meaning of words because they represent words, or n-grams, discreetly.

2.6 Medical Ontologies and Taxonomies

Medical ontologies are used in production systems as well as in research for medical data mining, on a very basic level systems check whether words that appear in the ontology are present in the text. Usually, you set a threshold to the number of matches. There are standard medical ontologies as well as customized ones for different purposes.

Medical terminology is a specific set of terms that are used in medicine or a sub-domain, subject, or specialty [50]. It can be divided into:

- A medical taxonomy - is a controlled vocabulary that has a hierarchy, it has an is-a relationship between terms, most of the medical terminologies follow into this category.
- A medical thesaurus - contains additional information, ex. Aspirin treats heart disease.
- Medical ontologies are mostly man-made, it is built to be used computationally.

Medical ontologies and taxonomies have extensively been used in research mainly for feature extraction in text classification for ADR and drug ADR relationship extraction. We will be summarizing few most used medical ontologies and relevant research that has used them.

2.6.1 UMLS (Unified Medical Language System)

UMLS is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems [51]. Common uses of UMLS includes,

- Link terms or codes between different entities such as doctor's prescription and insurance companies.
- Process texts to extract concepts, relationships, or knowledge.
- Develop medical information retrieval system.

UMLS has three sources of knowledge, namely,

- Meta thesaurus – Terms and codes from different medical vocabularies, with their definitions, hierarchies, relations, and other attributes.
- Semantic Network – medical semantic categories and their semantic relationships
- Specialist Lexicon and Lexical tools – A large syntactic lexicon of biomedical and general English with tools for generating lexical variants, normalizing Strings, and creating indexes.

Kathy et al. [12] have used sentences from the UMLS Metathesaurus to build an unlabeled dataset, they use this dataset to train a neural network model that learn text region embeddings representing high-level medical concepts, Karen et al. and Ning et al. [24,41] have used UMLS Metathesaurus for concept identification.

2.6.2 SNOMED-CT (Systematized Nomenclature of Medicine - Clinical Terms)

SNOMED CT is a standardized and multilingual vocabulary of clinical terms used by doctors and other health care professionals for the exchange of information. It contains approximately 300,000 medical concepts divided into hierarchies. Each has a unique number and more than one can be used to describe a medical condition [52]. Leaman et al. [53,54] have used SNOMED CT to create a vocabulary using medical concepts used for describing ADRs.

2.6.3 MedDRA (Medical Dictionary for Regulatory Activities)

MedDRA was developed by the ICH (International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human). It's a standardized medical terminology used for sharing information regarding medical regulatory information for medical products used by humans. Its coverage includes pharmaceutical, biologics, drug-device combination, and vaccine products [55]. [56] has used MedDRA codes as features for binary classification of text.

2.7 Techniques for ADR Detection

This section summarizes the different methods used for extracting ADR mentions from texts that possibly contain mentions of ADRs. Pharmaceutical companies and regulatory authorities are interested in the drug name and the specific ADR the drug has caused; the sections below look at different ways to extract these from ADR-containing texts.

2.7.1 Supervised Text Classification methods

Supervised learning algorithms such as SVM, NB, and ME have been used extensively in text classification. Here a manually annotated data set with ADR and non-ADR labels are given as a training set, based on the features we extract from this dataset the supervised classifiers learn and build a supervised model and use this model to predict future text segments. Research has used these supervised methods for binary classification of text into ADR and non-ADR classes [15,19].

When creating training data for ADR and Non-ADR classes data imbalance problem can arise because we cannot find ADR related text segments as much as Non-ADR related ones, specifically in social media datasets, Abeed et al. [15] have combined training data from different corpora as a solution to this problem. Also, ADR's for different drugs can be unique to those, hence our training dataset should be representative of all these types.

Abeed et al.[20] uses the supervised classification for detecting prescription medication overdose, it is said as the fastest-growing drug-related problem in the USA. They have collected tweets for

three of the most commonly abused drugs Adderall, oxycodone, and quetiapine, and manually evaluated these tweets whether each indicates the abusive use of a drug or not, they have also used a phonetic spelling generator to generate common misspellings that can occur in tweets. This data was used as the training set. The features they have used include word-n-grams, synonym expansions, whether the text contains abuse indicating terms, and drug slang. For the classification, they have used Naive Bayes, Support Vector Machines (SVMs), Maximum Entropy (ME), and a decision tree-based classifier(J48) algorithm.

CRF (Conditional Random Fields) is another classifier for structured prediction, it considers the context, it is used to label sequential data, for example, it is used for POS tagging in NLP [57]. Azadeh et al. [47] uses CRF to classify ADR tokens, they used word context features, POS tags, and word embeddings, for the word embeddings they used the word2vec tool.

2.7.2 Neural Networks

Supervised methods were used for text classification for a long time, but neural networks have shown promising results in this area lately, Siwei et al. [59] have used recurrent convolutional neural networks to classify text, they claim that their approach has less noise due to the capturing of contextual information with the word embeddings and they are also able to identify key terms in a text which play an important role in the whole text. Ji et al. [60] used RNN's (Recurrent Neural Network) and CNN's (Convolutional Neural Network) to classify short text nuggets for dialog act prediction.

Research done by Kathy et al. focused on using semi-supervised convolutional neural networks for classifying tweets with ADE mentions [12], they built several CNN models representing different types of data. They use a semi-supervised CNN framework that learns text region embeddings from the text. Using these learned embeddings and a manually annotated dataset another CNN was trained to form an ADE classifier. The research claims that LSTM's (Long Short-Term Memory) too perform very well too when doing text classification, but the improvements have been within 1% of the error rates when compared with CNN's, hence they leave LSTM as future work.

Xiao et al. [13] used a 4-layer CNN model consisting of a convolutional layer, max-pooling layer, dropout layer, and a fully connected output layer to classify election-related tweets, they have used word embeddings as features.

Shiyang et al [14] also have used a CNN model to detect the sentiment of entities such as products and events using Twitter. They have used a drop-out layer as a means of regularization of the network and an l2 norm constraint as a method to reduce overfitting.

2.8 Classification Algorithms

2.8.1 Logistic Regression

Logistic regression is a commonly used regression model, its features include.

- Used for binary & multiclass classification problems.
- With more data the accuracy increases.
- Do not need normal variable distribution.

2.8.2 Support Vector Machines

Support Vector machines have been extensively used in the field of natural language processing as shown in the literature review, its features include

- Mainly used for binary classifications but can be used for multiclass classifications using the One Vs Rest Strategy.
- Different kernels can be selected depending on the purpose
- Works with unstructured & semi-structured data.

2.8.3 Long Short-Term Memory (LSTM)

LSTM is a special kind of RNN (Recurrent Neural Network), they are capable of learning long-term dependencies which is a fundamental problem in RNN's, LSTM's can remember information for a long time.

2.8.4 Convolutional Neural Networks

CNN's are an artificial feed-forward neural network that is extensively been used for image classification tasks, but as pointed out in the literature review CNN's have shown promising results in the field of text classification, hence we experimented with it too.

CNN's are good with capturing the local dependencies such as in Image, Text, and time-series data. But CNN needs better computing power and a large amount of training data for improved accuracy.

2.9 Evaluation

The widely used evaluation methodologies for research are Precision, Recall, Accuracy, and F1. These may be used because of their simplicity and the accurate level of assessment they provide.

2.9.1 Precision

Out of the total set of positively identified content by the model, the fraction is relevant.

$$\text{Precision} = \frac{\text{Relevant Content} \cap \text{Retrieved Content}}{\text{Retrieved Content}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Equation 2.1: Precision Calculation

2.9.2 Recall

Out of the total positive content, the fraction was accurately identified by the model.

$$\text{Recall} = \frac{\text{Relevant Content} \cap \text{Retrieved Content}}{\text{Relevant Content}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Equation 2.2: Recall Calculation

2.9.3 Accuracy

The proportion of correctly classified content against incorrectly classified content.

$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{True Positives + True Negatives + False Positives + False Negatives}}$$

Equation 2.3: Accuracy Calculation

2.9.4 F1 Measure

F1 is a measure of accuracy derived using precision and recall.

$$F1 = \frac{2\,(\text{Precision.Recall})}{\text{Precison+Recall}}$$

Equation 2.4: F1 Calculation

2.10 Discussion

According to the discussion above, there has been researching focusing on different requirements, some for a specific set of drugs, some for a certain disease area, etc. The rule-based methods as well as the methods using medical ontologies and dictionaries need constant updating and they are suitable when we are interested in a specific category. Most research has used NLP-based features for Machine Learning classification, using methods such as word N-grams have given higher accuracies.

In the review, there was only one research that used a Neural Network which shows that majority is still relying on other methods such as NB and SVM algorithms. Although algorithms like SVM have proven improved accuracies for text processing neural networks tend to perform better as shown in the literature review.

Many types of research do not only depend on Twitter data, this may be due to the amount of data that can be obtained from Twitter which contains Drug Adverse Events, most of them have used data in combination with other sources. But they have used the same NLP methodologies for all these data from various sources, since twitter data is short content, and it contains entities such as hashtags, processing it using NLP methods specifically built for Twitter will aid in better identification of Adverse Events from these texts. Also, all the aforementioned research trains a model once, but in a production scenario that will be running continuously social media will be updated, hence the trained model will not be able to identify the new words and the slang used in Social media.

CHAPTER 3: METHODOLOGY

As we saw in the literature review, many approaches have been used by different researchers to automatically find adverse events from different sources, in this context capturing all the possible drug ADR's along with a few Non-ADRs captured as ADR's is acceptable than missing actual ADR's, hence precision is more important than recall.

The subproblems related to achieving the main target are as follows and will be discussing the approach to solve these in this chapter:

- Collection of social media content
- Data pre-processing techniques
- The possible features to extract
- Suitable methodologies for classifying ADR's
- Prepare the testing data and carrying out testing

Figure 3.1 depicts the sequence in which these tasks are carried out.

3.1 Proposed Architecture

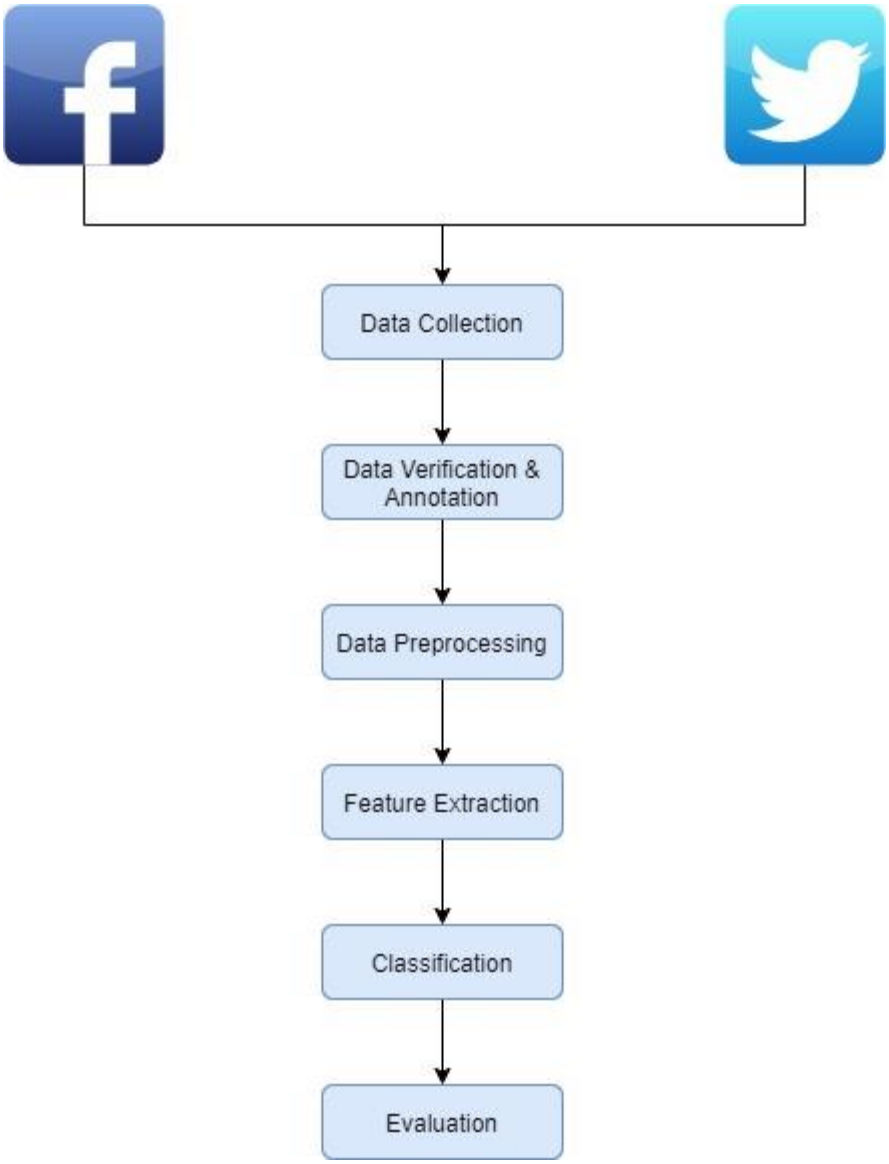

Figure 3.1 Proposed System Architecture

3.2 Data Collection

All of the research we evaluated in the literature review section has obtained their data directly from Social media using different means, these included using the social media API's and programmatically pulling data, some used publicly available datasets from other research and universities. The data from the freely available social media APIs has limitations and also the relevancy of the content to the keywords we supply is very low, hence a lot of advertisements and irrelevant content is received, previous research had to implement methodologies to filter these out. We can get highly relevant social media data from paid APIs such as the grip API [61] from Twitter but these are very expensive and hence not used for research.

We already had a dataset that was obtained from IQVIA[9]. This data was collected from their AETracker tool which gets social media data from their respective API's and also paid API's such as Gnip [61] and use a medical ontology to find the possible drug adverse event mentions, these were populated on to a dashboard where health care professionals review it and report back to the companies of those particular drugs. The Social Media APIs give us the option to follow keywords of interest and get relevant Social Media postings containing these keywords. Usually, social media data contains a lot of advertisements and marketing-related content, but luckily, we did not have this burden since the dataset was already filtered out for junk content through AETracker.

But the dataset from IQVIA [9] had data from Health care forums such as DailyStrength too since the language used in these differ from the language used in social media, we removed these as the accuracy of the trained model would drop.

Also, the AETracker tool was programmed according to the client definition of ADR for their drug, for example, one of the specifications were if there is the word "death" or a synonym of it, it would be considered an adverse event as death is considered as a serious ADR by definition, but this would not be the case always, for example:

"*I died laughing when I saw him gulping down 2 #*******"

Hence, we had to re-evaluate the dataset and mark the ones with clear ADR mentions only as positive. For this, we used two of our colleagues from IQVIA [9] who were health care professionals. The procedure for annotations was as follows:

- If there is a clear relationship between the ADR and the drug mentioned mark it as 1.
- If there is a drug and an ADR mentioned but not sure whether they are related, keep it as an UNDEFINED.
- If there are no ADRs and drug names mentioned or if there are, but you are confident that they are not related mark them as 0.

Using the above guideline, we re-annotated approximately 45000 tweets, Facebook posts, and comments.

For training the machine learning model we only selected the 1 and 0 content, we were left with 43073 social media posts. Some examples of these are given below:

- *I like ***** but it messes with my asthma* – 1
- *Might have to go get me some ****** before bed so you get zapped You big meanie* – 0
- *I've been taking it for a few days now. It does not seem to be making me better. Now my wife and son are sick. Any suggestions* – Undefined

Most of the previous research that we referred did not mention any consideration of this undefined category during their annotation, it is critical since for it to become an adverse effect there should be clear evidence that it was caused by the drug. Also, we had a fairly balanced dataset of ADR's and Non-ADR's, most of the previous research struggled to achieve this balance on the training set, we managed it since we obtained the data from AETracker which has been under operation since 2012 and there was plenty of Social media data in its databases.

3.3 Inter-rater Agreement

Table 3.1 lists the inter-rater agreement for the dataset. Equation 3.1 shows the calculation of Kohen's Kappa measure. 500 posts, comments, and tweets were analyzed by both annotators for the calculation.

		Annotator 1		
		ADR	Non-ADR	Undefined
Annotator 2	ADR	232	2	0
	Non-ADR	3	236	4
	Undefined	0	3	20

Table 3.1: Inter-rater Agreement

$$Kappa(k) = \frac{\mathrm{P}_0 - \mathrm{Pe}}{1 - \mathrm{Pe}} = \frac{0.97 - 0.458}{1 - 0.458} = 0.945$$

Equation 3.1: Kohen's – Kappa

As you can see from the above result we have a near-perfect agreement, this is because there is a clear guideline for the text to become an ADR mention, and also both the annotators were pharma co-vigilance analysts and they have been doing this as part of their job for a long time.

3.4 Data Preprocessing

Preprocessing plays an important role in the accuracy of the classification task, unlike author-written documents such as newspaper articles text in social media does not contain formal writing language, it includes a lot of slang, punctuation marks, hashtags, and usernames mentions. All these contribute to the meaning of the text, hence cannot be removed, out of the below-preprocessing methodologies we try to apply ones that apply to social media content.

- Remove numbers
- Remove punctuations
- Removing extra white spaces and line breaks
- Remove stop-words

- Remove sparse terms
- Stemming
- Lemmatization
- Spell checking
- Removing username and hashtags

Many of the above-preprocessing steps have been carried in previous research which used Twitter and Facebook data [24,12,58,62].

Although removing numbers and punctuations was easy using regular expressions, we did not remove them as they contribute to the meaning in social media texts and some generic drug names include numbers.

All extra white spaces and line breaks were removed, there were a lot of these as we got database dumps from the AETracker tool, these were removed using regular expressions.

We did experiments with both including stop words and excluding stop words since some research in this field claimed that removing stop words for social media text reduces the accuracy, we used the standard set of English stop words from the Python NLTK[63] package.

Removing sparse terms was not considered because healthcare-related terms such as drug names and disease conditions can be there in the corpus which is not frequent, dropping these would miss out on possible ADRs.

We tested out both stemming and lemmatization using Porter Stemmer [64] and the wordnet lemmatizer because some research claimed that these improved accuracies but some did not use these at all, this can be because Social media does not use formal written language and has a lot of slang usage.

Spell correction was not tried out as Social media conversations contain a lot of slang and a lot of short forms in words, correcting these automatically would alter the meaning.

We removed Twitter handle mentions and Facebook username mentions which started with the @ symbol using regular expressions since they did not add any meaning to the content, but we did not remove hashtags as health condition mentions and drug names appear in the form of hashtags. For example, #cancer, #melanoma and #diabetes.

Previous research in this area had tried different preprocessing steps, most of them removed punctuation marks, @ symbols, hashtags, and did spell correction. Stemming and lemmatization were also extensively used as a preprocessing step.

3.5 Feature Selection

Now we should convert the preprocessed social media content into a form that a machine learning algorithm could understand, in other words, we should extract a set of features. There are many features that we can employ for a text classification process as shown in the literature review.

- Bag of Words model
- Word N-grams
- TF-IDF (Term Frequency Inverse Document Frequency)
- POS Tagging
- Change Phrases
- Word embeddings

Bag of words is a simple commonly used model in text classification, it creates a set of all words present in the entire corpus and the presence of each word is used as a feature, it disregards positional information and the grammatical structure. Word N-grams is an extension to this where it considers sets of words as features, for example, Bigrams (n=2) considers sets of two consecutive words, and the presence of the words in that order is considered as a feature, thus positional information is considered. Previous research in this area has not used this feature much since there are more advanced and accurate features that we can use.

Different words will have varying importance based on the context, common words such as prepositions would become frequent in a model such as a bag of words, but these words carry less value with regards to the experiment, word weighting schemes can be introduced to diminish the importance of these common words, TF-IDF considers the frequency on which a certain word occurs in a document (Term Frequency), to negate the effect of common words it adds inverse document frequency.

POS tagging is a methodology where we tag the parts of speech in the corpus text and use this as features for the machine learning model, POS Tags has not been used much in previous research work in this area, one reason for this is that social media contains a lot of slang and informal text. [17] has developed a state-of-the-art POS Tagger specifically for Twitter and IRC (Internet relay chats) using a POS tagger from another research.

Word Embeddings is a promising recent development in the field of natural language processing. It transforms the one-dimensional word into a much denser vector space, in this transformation it maps the similar words near each other in the vector representation.
Word2vec is the most popular form of word embeddings in use today, it employs deep learning to learn a word embedding model.
Change Phrases has been used in research in the past, the intuition is that we can determine whether a sentence represents positive or negative information by observing the change, if a bad thing (example headache) was reduced then it is a positive outcome whereas if it increases it is a negative outcome. This feature tries to track whether a good or bad thing increases or decreases. Sarker et. al [15] has used this feature in their research, they use words captured by previous research and check whether either of these good or bad words occur around a specified window of the word "MORE" or "LESS", based on whether the word occurring belonged to a good or bad category they activate the change and use it as a feature.

We did several experiments using word2vec, first, we generated a word2vec model using our corpus, for this we used the genism [47] python library. Word2Vec can be generated using the skip-gram or CBOW (Continuous Bag of Words) model, skip-gram model learns to predict the nearby words to the given input while the CBOW model tries to predict a word given its surrounding words.

Below are the different configurations that we used to train the word2vec model.

Property	Values Tested
Dimensions	50, 100, 200, 300, 400
Word min count	1, 3, 5
Window Size	5,10

Table 3.2: Word2Vec Attributes

In the above properties dimensions is the dimension of the vectors of the word2vec model, out of the above values a dimension of 300 gave us the maximum accuracy. The minimum word count is the number of times a word should appear for that to be included in the model, since we had a small corpus there was less chance a word appears several times, hence 1 for this parameter gave us the most accuracy. Window size is the number of surrounding words to be considered, again since we had a small corpus, we went with the value of 5.

We also tested two widely used prebuilt word vector models, Glove [65] and Google News Vectors. We used the Twitter Glove vectors which were trained on a corpus of 6 billion words, we tested both with the 50 dimensions and 300 dimensions, we specifically went with 300 dimensions since previous research has used that dimensionality and achieved good results. Google news vectors have been trained on 3 billion words from different news articles, its dimensionality is 300.

The final testing was using the python keras [66] embedding layer, we used this as the first layer of our deep learning model, it creates a dense vector from the given input integers. We used 300 as our dimensionality here too. But for us to use the embedding layer we should convert our textual training data into an integer representation, for this, we used the keras [66] tokenizer, which converts each line into a vector of integers, each of these integers is indexes of a dictionary built from the training corpus. For these vectors to be of the same length we pad the sequences at the end with zeroes.

Previous research in this area did not consider prebuilt word embeddings such as Glove [65] and Google news vectors, their main research aim was to test the effectiveness of the background corpora of the word embedding model on the accuracy, for that they trained their word embeddings models. Glove specifically has a model trained on tweets that suit our purpose well. In our research, we use the genism [47] library in a similar way to train our own word embeddings model using health-related Twitter data.

3.6 Classification algorithms

The next step is selecting the classification methodology to classify social media content using the above-mentioned features. We experimented with few machine learning algorithms which are listed below.

- Logistic Regression
- Support Vector Machines
- Convolutional Neural Networks (CNN)
- Long Short-Term Memory Networks (LSTM)

3.6.1 Logistic regression

We used the Logistic Regression implementation of the python Scikit Learn[62] library. There are different parameters which this library allows to tune such as inverse regularization which reduces overfitting.

Previous research in this area has not used Logistic Regression for their experiments, but it has been used widely for sentiment classification, hence we tried to test the accuracy of it for our research area. Logistic Regression being well known for binary classification is also another reason we chose to experiment with it.

3.6.2 Support Vector Machines

There are different parameters Scikit learns [62] implementation of SVM allows to tune, from those we experimented with different values for the penalty for error parameter. SVM has been widely used for text classification tasks in general and sentiment classification specifically.

3.6.3 Convolutional Neural Networks

We used the python keras [66] with the Tensorflow backend implementation of CNN. We tried different combinations of layers and within the layers, we experimented with different parameters given below in the table.

Layer	**Parameter**	**Values**	**Description**
Embedding	Vocabulary Size	The actual size of training vocabulary	The number of words from the corpus we consider building the embedding model.
Embedding	Embedding Dimension	50, 100, 200, 300, 400	The dimension of the vectors in the model built.
Dropout	Rate	0.1, 0.2, 0.5	Drops out nodes during each epoch at the given rate to prevent overfitting.
Conv1D	Filters	64,128,256	Number of the output filters in the convolution.
Conv1D	Kernel size	5	Specifies the length of the 1D convolution window.
Dense	Activation	relu	The activation function for the hidden layers.

Output	Activation	Sigmoid, linear	The activation function of the output layer, sigmoid was more suitable since we need a binary output.
Output	Activity Regularize	l2(0.0005)	Regularization is applied to the activation of the output layers.

Table 3.3: CNN Parameters

As the loss function for the network, we experimented with binary cross-entropy and categorical hinge. When the activation function of the output layer was linear categorical hinge performed better but the activation function as sigmoid binary cross-entropy performed better. The optimizer for the network was used as adam.

Some of the previous research in this area has used CNN's for text classification. Kathy et. al [12] used a CNN model, but it was an already built model from another research hence they did not experiment with the layer configuration or parameters such as regularization or methods to prevent over-fitting of data which is common in Neural networks. Xiao et. al [13] used CNN and word embeddings for classifying tweets, they experimented with different configurations of the CNN but did not try out parameters such as activity and weight regularization. But we have tried out the above parameters with different layers of CNN to improve the performance.

3.5.4 Long Short-Term Memory (LSTM)

We used the python keras [66] with the Tensorflow backend implementation of the LSTM layer, word embeddings were used as the feature for this too. Below were the parameters for the LSTM layer that we experimented with.

Layer	Parameter	Values	Description
LSTM	Units	100, 256	The dimensionality of the output space.
LSTM	Activation	relu	The activation function for the layer.
SpatialDropout	Rate	0.1, 0.2	Dropout entire feature maps from the convolutional layer which is then not used during pooling.
LSTM	dropout	0.1, 0.2	Drops out nodes during each epoch at the given rate to prevent overfitting.
LSTM	Recurrent dropout	0.1, 0.2	Masks the connections between recurrent units.

Table 3.4: LSTM Parameters

Kathy et. al [12] shows that LSTM is a promising area for experimenting with text classification, but none of the previous research in this area experimented with any RNN type networks such as LSTM's. We experimented with different parameters in LSTMs as shown above.

CHAPTER 4: SYSTEM EVALUATION

This chapter explores the results of different experiments we did on classification algorithms, features, and preprocessing techniques and compares the results with the existing research.

The goal of the research is to classify drug adverse events effectively from Social media content, the task includes cleaning the data appropriately, extracting features, and selecting a suitable classification algorithm. The accuracy of the classification depends on these three tasks and each of these tasks we did is compared to the existing research.

4.1 Baseline experiments

The base experiment that we are comparing with is research done by Kathy et al. [12] on using artificial Neural Networks to detect drug adverse events using Tweets, since most of our data is also tweets the results can be compared, hence we use this as the main research that we will be using as the baseline, but they have only experimented with CNN models using phrase embeddings trained on different corpora, hence to show the accuracy of other algorithms will be using another research done in this area [4], Table 4.1 summarizes there results.

Algorithm	**Accuracy**	**F-Score**
SVM	86.2%	53.8%
CNN	NA	64.50%

Table 4.1: Baseline experiment results

Without any preprocessing of the data such as removing stop words, stemming, or lemmatization we tested SVM, Logistic Regression, CNN, and LSTM algorithms, table 4.2 summarizes the results.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-Score**
SVM	0.87	0.87	0.87	0.87
Logistic Regression	0.87	0.87	0.87	0.87
CNN	0.87	0.87	0.86	0.86
LSTM	0.56	0.57	0.55	0.55

Table 4.2: Initial experiment results

Considering the F-score SVM, Logistic Regression and CNN gave us the topmost results as shown. The fact that CNN and LSTM not showing accuracies better than SVM and Logistic regression can be due to insufficient training data, usually, deep learning methodologies perform well with a lot of training data.

4.2 Preprocessing

As mentioned in the methodology section we did not try removing punctuation marks since social media has short texts and the punctuation marks add meaning to these. But we experimented with and without stop words, lemmatization, and stemming.

4.2.1 Stop words

Algorithm	Accuracy	Precision	Recall	F-Score
SVM	0.86	0.86	0.86	0.86
Logistic Regression	0.86	0.86	0.86	0.86
CNN	0.86	0.86	0.85	0.86
LSTM	0.51	0.52	0.50	0.50

Table 4.3: With Stop Words

Table 4.3 gives the results of the four algorithms without the removal of stop words, as you can see the performance has slightly dropped compared to it with stop words. Though research claims that stop words do not add any meaning removal of those will increase the accuracies, surprisingly it has dropped in our case.

4.2.2 Lemmatization

Algorithm	Accuracy	Precision	Recall	F-Score
SVM	0.88	0.88	0.88	0.88
Logistic Regression	0.88	0.88	0.88	0.88
CNN	0.85	0.84	0.86	0.85
LSTM	0.56	0.57	0.55	0.55

Table 4.4: Lemmatization

Table 4.4 summarizes the results of the four algorithms with lemmatization as the only preprocessing. The values indicate that lemmatization slightly increases the accuracy of the SVM and the Logistic Regression models, but it drops the performance of the CNN. Previous research that used non-deep learning models for text classification also claims that lemmatization improves accuracy, and it is depicted here too.

4.2.3 Stemming

Algorithm	Accuracy	Precision	Recall	F-Score
SVM	0.87	0.87	0.87	0.87
Logistic Regression	0.87	0.87	0.87	0.87
CNN	0.87	0.87	0.86	0.86
LSTM	0.55	0.55	0.55	0.55

Table 4.5: Stemming

As shown in table 4.5 stemming does not improve the accuracy as much as it does with lemmatization, but comparatively stemming has a performance improvement on CNN which we did not see with lemmatization.

4.2.4 Stemmed and Lemmatized

Algorithm	Accuracy	Precision	Recall	F-Score
SVM	0.88	0.88	0.88	0.88
Logistic Regression	0.88	0.88	0.88	0.88
CNN	0.85	0.84	0.86	0.85
LSTM	0.59	0.60	0.57	0.58

Table 4.6: Stemmed and Lemmatized

With stemming and lemmatization both together there is no improvement above 88% in SVM and Logistic Regression which is the same performance as we got with only lemmatization, it also drops the performance of the CNN model. Hence, we are going to continue the rest of our experimentation with lemmatization as our preprocessing step. Table 4.7 summarizes the results for the algorithms with all the preprocessing steps combined.

Algorithm	Accuracy	Precision	Recall	F-Score
SVM	0.88	0.88	0.88	0.88
Logistic Regression	0.88	0.88	0.88	0.88
CNN	0.85	0.84	0.86	0.85
LSTM	0.59	0.60	0.57	0.58

Table 4.7: All Preprocessing Combined

4.3 Features

In this section, we will show the results of using selected features against each machine learning algorithm. From the section before we saw that lemmatization gives the highest performance for SVM and Logistic regression while stemming works the best for CNN. Since the features highly influence the performance of the algorithm we will be experimenting with all feature and algorithm combinations. Following is the list of features that are to be evaluated in this section.

- Word-2 grams
- TF-IDF
- Word Embeddings Prebuilt (word2vec, Glove)
- Word vectors built with our corpus
- Keras word embedding layer

We tested word-2 grams and TF-IDF with SVM and Logistic Regression (LR) algorithms, table 4.8 summarizes the results of those tests.

Algorithm	Feature	Accuracy	Precision	Recall	F-Score
SVM	Word-2 grams	0.8556	0.8569	0.8556	0.8545
SVM	TF-IDF	0.8906	0.8919	0.8906	0.8900
LR	Word-2 grams	0.8612	0.8645	0.8612	0.8598
LR	TF-IDF	0.8839	0.8850	0.8839	0.8832

Table 4.8: Features with Algorithms

4.3.1 Word Embeddings

4.3.1.1 Glove and Google News Vectors

We will first use Glove Twitter vectors and Google News Vectors prebuilt word embeddings with the CNN model, here we experimented with 300-dimension versions of each of these vectors since it was the most used dimension in previous research.

Model	Accuracy	Precision	Recall	F-Score
Glove	0.8552	0.8642	0.8550	0.8490
Google News Vectors	0.8355	0.8365	0.8244	0.8230

Table 4.9: Glove and Google News Vectors

As you can see in Table 4.9 Glove is slightly accurate over Google News vectors, this is since we used the Glove Twitter vectors, this shows the effect of the background corpora on the performance of the classification.

4.3.1.2 Vector Dimension

We also tested how the size of the word vector affects the word representation in our corpus, for this test we used Glove [65], we could not get the Google Negative News Vectors model in different dimensions hence it could not be tested out for word representations.

Vector Size (Dimensionality)	Word Representation
50	0.42
100	0.58
200	0.59
300	0.75

Table 4.10: Glove Word Representation

4.3.1.3 Word Vectors built from the background corpus

We used the python genism [47] library to generate word2vec vectors from our dataset, we used 300 as the dimensionality for the vectors. Before showing the effectiveness of this on the classification we will show the most relative words and their similarity values for some of the common words in our dataset such as “mucus” and “cough”

Keyword	Similarity
congestion	0.73
mucus	0.60
coughing	0.57
cold	0.57
nose	0.57
seizures	0.51
sinus	0.49
nastiest	0.48
loosen	0.47
fits	0.46

Table 4.11: Most similar words for “cough”

Keyword	Similarity
cough	0.60
nose	0.55
congestion	0.55
dry	0.55
phlegm	0.54
crud	0.52
stuff	0.51
chest	0.50
mucous	0.50
weather	0.49

Table 4.12: Most similar words for "mucus"

From this we see that it was effectively able to capture words with a similar context, the most similar words have a higher similarity value, this is based on the usage of these words in the corpus we used. You can also see that it was able to capture misspelled words e.g. mucous for mucus, this is common in social media content. By increasing the size of the training dataset, we can further improve the accuracy of the word2vec model.

Next, we will see the accuracy of this model with CNN and LSTM.

Model	Accuracy	Precision	Recall	F-Score
CNN	0.8640	0.8642	0.8550	0.8558
LSTM	0.6012	0.6124	0.6044	0.5992

Table 4.13: CNN with Gensims

As you can see in table 4.13 using a word2vec corpus built from the background corpus gives a slight improvement in the LSTM, but it is not comparable with the accuracies already achieved by CNN.

4.3.1.4 Embedding layer

We tested the Keras [66] word embeddings layer as features for the CNN and LSTM models, we tested the same vector sizes we used for testing word representation above on the Keras [66] embedding layer, the below table lists the outputs.

Algorithm	Vector Size (Dimensionality)	Accuracy	Precision	Recall	F-Score
CNN	50	0.6232	0.6342	0.6231	0.6142
CNN	100	0.8255	0.8365	0.8244	0.8230
CNN	200	0.8565	0.8565	0.8570	0.8540
CNN	300	0.8592	0.8597	0.8585	0.8540
LSTM	50	0.4240	0.4232	0.4432	0.4100
LSTM	100	0.4840	0.4931	0.4800	0.4800
LSTM	200	0.5800	0.5878	0.5800	0.5793
LSTM	300	0.5962	0.6012	0.5900	0.5911

Table 4.14: Embedding Layer

4.4 Classification Algorithms – Hyper Parameter Tuning

In this section we will look at the performance of different machine learning classifiers that we tested in this research; we will be parameter tuning different algorithms to check the performance of those.

4.4.1 CNN

From the previous chapter we saw that the CNN model behaves the best with the word embeddings layer on Keras [66], hence first we will try to tune different parameters and add different layers to the CNN model to check the accuracy.

Conv1D Filter

Value	Accuracy	Precision	Recall	F-Score
128	0.8534	0.8565	0.8531	0.8520
256	0.8420	0.8419	0.8431	0.8410

Table 4.15: Conv1D Filter

Dropout Regularize

Value	Accuracy	Precision	Recall	F-Score
0.2	0.8561	0.8565	0.8531	0.8540
0.1	0.8422	0.8419	0.8422	0.8410

Table 4.16: Dropout Regularize

Activity Regularization

Value	Accuracy	Precision	Recall	F-Score
l2(0.00005)	0.8588	0.8590	0.8589	0.8571
l2(0.00003)	0.8511	0.8532	0.8410	0.8500

Table 4.17: Activity Regularize

As you can see there are minor improvements when tuning hyperparameters of the CNN. Looking at the results we get the maximum performance of the CNN model with a drop-out layer of 0.2 and l2 regularizing the output layer with a value of 0.00005.

4.4.2 LSTM

We will next try to tune different parameters and add different layers to the LSTM model to check the accuracy.

Spatial Dropout layer

Value	Accuracy	Precision	Recall	F-Score
0.1	0.5800	0.5878	0.5800	0.5793
0.2	0.5832	0.5962	0.6012	0.5900

Table 4.18: Spatial Dropout Layer

Dropout Regularize

Value	Accuracy	Precision	Recall	F-Score
0.2	0.5822	0.5800	0.5888	0.5811
0.1	0.5821	0.5800	0.5887	0.5811

Table 4.19: LSTM Dropout Regularize

Recurrent Dropout

Value	Accuracy	Precision	Recall	F-Score
0.2	0.5822	0.5800	0.5888	0.5811
0.1	0.5821	0.5800	0.5887	0.5811

Table 4.20: Recurrent Dropout

4.4.3 SVM

Table 4.21 shows the metrics while changing the c value of the SVM model. We only tested with the 'Linear' kernel; previous research suggested that this kernel best works for text classification tasks. We experimented with different values for the inverse of regularization strength (c), this value defaults to 1.0 if not specified explicitly.

The Inverse of Regularization Strength

Value	Accuracy	Precision	Recall	F-Score
0.5	0.8887	0.8911	0.8887	0.8879
2	0.8867	0.8876	0.8867	0.8861
3	0.8883	0.8888	0.8883	0.8878

Table 4.21: SVM Inverse of Regularization Strength

4.4.4 Logistic Regression

Table 4.22 lists the results obtained while changing the inverse of regularization strength (c) of the Logistic Regression model. This value defaults to 1.0 if not specified explicitly.

The Inverse of Regularization Strength

Value	Accuracy	Precision	Recall	F-Score
0.5	0.8803	0.8816	0.8803	0.8795
2	0.8867	0.8876	0.8867	0.8861
3	0.8861	0.8869	0.8861	0.8855

Table 4.22: LR Inverse of Regularization Strength

4.5 Confusion Matrices

We will see the confusion matrix for each of the three algorithms with their best-performing hyperparameters.

		ADR	Non-ADR
Actual	ADR	3114	687
	Non-ADR	344	4470

Fig 4.1: Confusion matrix for Logistic Regression

		ADR	Non-ADR
Actual	ADR	3118	683
	Non-ADR	275	4539

Fig 4.2: Confusion matrix for SVM

		ADR	Non-ADR
Actual	ADR	3100	669
	Non-ADR	275	4539

Fig 4.3: Confusion matrix for CNN

As you can see from the confusion matrices SVM has performed the best out of all the three algorithms with the least false positives and least false negatives.

4.6 Error Analysis

In this subsection, we briefly examine few misclassified instances. Table 4.23 lists down few misclassified instances. We have removed the names of drugs wherever they are mentioned. Label 1 depicts an ADR while 0 for Non-ADR.

Index	Comment	Label	Prediction
1	Used some of this stuff on my neighbor's kid guess he's the 001 Wish you guys could figure that out	1	0
2	Well Lysol much like your ads of decades past this one isn't cutting it for me	1	0
3	your period will be crazy for about three months after a ****** removal or at least mine was also not to scare you but the ****** is what has caused some serious feility issues for me as far as ovulation and premature eggs its rare and i highly doubt that is the case for you or most women	1	0
4	sharing some groovy love	0	1
5	Santos Tovar	0	1

Table 4.23: Error Analysis

- Comment 1 has been predicted as Non-ADR, but it is an ADR because it implies that the neighbor's kid is indestructible by saying "001". The word "001" would rarely occur elsewhere.
- Comment 2 does not have any ADR-related terms or drug names which can be the reason why the classifier did not identify it.
- Comment 3 has drug name mentions and ADR mentions although "feility" is misspelled it does not have the mention of a direct relationship between the two though it implies.
- Comment 4 can be misclassified due to the presence of the adjective "groovy"
- Comment 5 is a name of a person who should rightfully be classified as Non-ADR.

4.7 SVM Vs LSTM

As you can see in the results SVM performs the deep learning techniques. SVM works by finding the hyperplane (margin) that best separates the two data classes. Since our problem is a binary classification this works very well.

In contrast, LSTM is a type of RNN which tries to learn a sequence in the dataset, it is good for datasets where a hierarchical decomposition may exist. But since ours is a binary classification problem there is hardly a sequence that the LSTM can learn from. This can be a reason why LSTM showed poor results compared to SVM.

4.8 Summary

From the error analysis section, we can conclude that the classifier is unable to perform well when there is no direct relationship between a drug name and an ADR or a mention of an ADR. It also misclassifies when there are spelling mistakes, but it is common in a social media text. Adding a spell correction mechanism specific to social media text and increasing the training corpus size with content other than health-related content so that it can capture implications like "001" for strength will increase the accuracy.

Before the conclusion of this section, we will show the best results of the three algorithms:

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-Score**
SVM	0.8887	0.8911	0.8887	0.8879
Logistic Regression	0.8803	0.8816	0.8803	0.8795
CNN	0.8596	0.8599	0.8521	0.8571

Table 4.24: Summary

CHAPTER 5: CONCLUSION

Finding and reporting ADR's at the earliest possible time after it occurs has been a challenge in the pharmaceutical industry, this research focused on developing a better ADR Classifier from social media content which would enable ADR's to be reported in a shorter period. We used a highly subject-oriented dataset from IQVIA [9] which contained 43000 odd tweets, it is the biggest dataset used in research in this area. We manually annotated it for machine learning with the support of two health care professionals and tagged them for the presence or absence of adverse events.

Preprocessing was mostly to remove unnecessary white spaces, symbols such as emojis which is common in social media and removing HTML tags. We did not remove any punctuation marks or stop words and we also saw that removing these does not contribute a lot to the performance. We lemmatized the content as a preprocessing step.

Common features such as TF-IDF, Word n-grams, and word embeddings were tested with 4 machine algorithms. Out of these features, word embeddings and TF-IDF showed promising results. SVM, Logistic Regression, and CNN performed the best for our dataset, although research suggested that LSTM's have promising results in the field of text classification, we were not able to prove that through our experiments.

We also experimented with different forms of word embeddings such as prebuilt word vectors, building word vectors using our corpus and the Keras [66] embedding layer. There were different parameters to tune in these such as the dimensionality of the vectors, minimum word counts, etc. We tested word embeddings with our CNN model and using the Keras embedding layer gave us the best results.

We tuned different parameters of our CNN model and tested different layer configurations, most of those were done to prevent overfitting which is a common problem in neural networks. Using dropout between deep layers and regularizing activity in certain layers gave us the best results.

We were able to obtain the best classification results so far in the area of mining drug adverse events from Social media content. We also showed how effective well-known text classification

algorithms such as SVM and Logistic Regression can be effective in the field of ADR mining. We experimented with CNN and different parameters of a CNN model and were able to show that it is very effective in text classification problems such as this.

5.1 Future Work

Following is a list of future work that can be carried out:

- Develop the model to extract the exact ADR from the social media text, if this can be done with very high accuracy the human interaction for this process can be eliminated, and the whole drug ADR reporting process can be automated.
- Experiment more on word embeddings as a feature and combining word embeddings and traditional features.
- Further improving preprocessing steps such as stemming, lemmatization would improve the accuracy further.
- The sentiment of words and change phrases can experiment as a feature for ADR mining.
- Spell correction related to social media content and using medical term corpora to spell correct medical-related terms would improve the accuracy further.
- Incorporating social media non-health-related content into the training corpus would enable it to capture things people are trying to say indirectly in conversations.
- Experimenting more with deep learning to track ADRs.

REFERENCES

1. https://www.pharmaceutical-journal.com/publications/tomorrows-pharmacist/drug-development-the-journey-of-a-medicine-from-lab-to-shelf/20068196.article?firstPass=false
2. Schatz, S.N., & Weber, R. (2018). Adverse drug reactions. British Medical Journal, 363.
3. https://www.ncbi.nlm.nih.gov/pubmed/16808549
4. https://www.who.int/medicines/areas/quality_safety/safety_efficacy/pharmvigi/en/
5. http://broughttolife.sciencemuseum.org.uk/broughttolife/themes/controversies/thalidomide
6. http://apps.who.int/medicinedocs/en/d/Js4893e/5.html
7. http://www.pharmexec.com/pharma-companies-can-solve-social-media-adverse-events-reporting-problem-and-stop-worrying
8. https://www.fda.gov/media/122835/download
9. https://www.iqvia.com/
10. https://www.iqvia.com/solutions/integrated-global-compliance/safety-and-pharmacovigilance
11. P Tafti, A., Badger, J., LaRose, E., Shirzadi, E., Mahnke, A., Mayer, J., Ye, Z., Page, D. and Peissig, P., 2017. Adverse Drug Event Discovery Using Biomedical Literature: A Big Data Neural Network Adventure. JMIR Medical Informatics, 5(4), p.e51.
12. Lee, K., Qadir, A., Hasan, S., Datla, V., Prakash, A., Liu, J. and Farri, O., 2017. Adverse Drug Event Detection in Tweets with Semi-Supervised Convolutional Neural Networks. Proceedings of the 26th International Conference on World Wide Web, pp.705-714.
13. Yang, X., Macdonald, C. and Ounis, I., 2017. Using word embeddings in Twitter election classification. Information Retrieval Journal, 21(2-3), pp.183-207.
14. Liao, S., Wang, J., Yu, R., Sato, K. and Cheng, Z., 2017. CNN for situations understanding based on sentiment analysis of twitter data. Procedia Computer Science, 111, pp.376-381.
15. Sarker, A. and Gonzalez, G., 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. Journal of Biomedical Informatics, 53, pp.196-207.

16. Owoputi, O. & O'Connor, B. & Dyer, C. & Gimpel, Kevin & Schneider, N. & Smith, N.A.. (2013). Improved part-of-speech tagging for online conversational text with word clusters. Proceedings of NAACL-HLT. 2013. pp.380-390.
17. https://nlp.stanford.edu/software/tagger.shtml
18. https://www.nlm.nih.gov/research/umls/
19. Ofoghi, Bahadorreza & Siddiqui, Samin & Verspoor, Karin. (2016). READ-BioMed-SS: Adverse drug reaction classification of microblogs using emotional and conceptual enrichment.
20. Sarker, A., Malone, D. and Gonzalez, G., 2017. Authors' Reply to Jouanjus and Colleagues' Comment on "Social Media Mining for Toxicovigilance: Automatic Monitoring of Prescription Medication Abuse from Twitter". Drug Safety, 40(2), pp.187-188.
21. Wolpert, D., 1992. Stacked generalization. Neural Networks, 5(2), pp.241-259.
22. Zhang, Z., & Nie, J. (2015). AN ENSEMBLE METHOD FOR BINARY CLASSIFICATION OF ADVERSE DRUG REACTIONS FROM SOCIAL MEDIA.
23. Jonnagaddala, J., Jue, T.R., & Dai, H. (2015). BINARY CLASSIFICATION OF TWITTER POSTS FOR ADVERSE DRUG REACTIONS.
24. O'Connor, K., Pimpalkhute, P., Nikfarjam, A., Ginn, R., Smith, K. L., & Gonzalez, G. (2014). Pharmacovigilance on twitter? Mining tweets for adverse drug reactions. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2014, 924–933.
25. Kandula, S., & Zeng-Treitler, Q. (2010). Exploring relations among semantic groups: a comparison of concept co-occurrence in biomedical sources. Studies in health technology and informatics, 160(Pt 2), 995–999.
26. Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F. and Salakoski, T., 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. BMC Bioinformatics, 9(S11).
27. Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F. and Salakoski, T., 2008. Comparative analysis of five protein-protein interaction corpora. BMC Bioinformatics, 9(S3).
28. Jang, H., Lim, J., Lim, J., Park, S., Lee, K. and Park, S., 2006. Finding the evidence for protein-protein interactions from PubMed abstracts. Bioinformatics, 22(14), pp.e220-e226.

29. Rinaldi, F., Schneider, G., Kaljurand, K., Hess, M., Andronis, C., Konstandi, O. and Persidis, A., 2007. Mining of relations between proteins over biomedical scientific literature using a deep-linguistic approach. Artificial Intelligence in Medicine, 39(2), pp.127-136.
30. Fundel, K., Kuffner, R. and Zimmer, R., 2006. RelEx--Relation extraction using dependency parse trees. Bioinformatics, 23(3), pp.365-371.
31. Kang, N., van Mulligen, E. and Kors, J., 2011. Comparing and combining chunkers of biomedical text. Journal of Biomedical Informatics, 44(2), pp.354-360.
32. HUANG, M., ZHU, X., & LI, M. (2006). A hybrid method for relation extraction from biomedical literature. International Journal Of Medical Informatics, 75(6), 443-455.
33. Buchholz S, Marsi E (2006) CoNLL-X shared task on multilingual dependency parsing. Proceedings of the Tenth Conference on Computational Natural Language Learning. Pp 149-164.
34. https://ctakes.apache.org/
35. Gurulingappa, H., Mateen-Rajput, A., & Toldo, L. (2012). Extraction of potential adverse drug events from medical case reports. Journal Of Biomedical Semantics, 3(1), 15.
36. Yang, Z., Lin, H., & Li, Y. (2010). BioPPISVMExtractor: A protein–protein interaction extractor for biomedical literature using SVM and rich feature sets. Journal Of Biomedical Informatics, 43(1), 88-96.
37. Bui, Q., Katrenko, S., & Sloot, P. (2010). A hybrid approach to extract protein-protein interactions. Bioinformatics, 27(2), 259-265.
38. D. Zelenko, C. Aone, A. Richardella. (2003) Kernel methods for relation extraction.The Journal Of Machine Learning Research.
39. R.C. Bunescu, R.J. Mooney [2005]. A shortest path dependency kernel for relation extraction.Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 724-731.
40. Schuemie, Martijn & Jelier, Rob & Kors, Jan & Nl, J. (2007). Peregrine: Lightweight gene name normalization by dictionary lookup. Proceedings of the Biocreative 2 workshop.
41. Kang, N., Singh, B., Bui, C., Afzal, Z., van Mulligen, E., & Kors, J. (2014). Knowledge-based extraction of adverse drug events from biomedical text. BMC Bioinformatics, 15(1).

42. Huang, Y. (2005). Improved Identification of Noun Phrases in Clinical Radiology Reports Using a High-Performance Statistical Natural Language Parser Augmented with the UMLS Specialist Lexicon. Journal of The American Medical Informatics Association, 12(3), 275-285.
43. Demner-Fushman, D., Chapman, W., & McDonald, C. (2009). What can natural language processing do for clinical decision support?. Journal Of Biomedical Informatics, 42(5), 760-772.
44. Huh, J., Yetisgen-Yildiz, M., & Pratt, W. (2013). Text classification for assisting moderators in online health communities. Journal Of Biomedical Informatics, 46(6), 998-1005.
45. http://wordnet.princeton.edu/
46. Niu, Y., Zhu, X., Li, J., & Hirst, G. (2005). Analysis of polarity information in medical text. AMIA . Annual Symposium proceedings. AMIA Symposium, 2005, 570–574.
47. https://radimrehurek.com/gensim/models/word2vec.html
48. Pancoast, S., & Akbacak, M. (2012). Bag-of-Audio-Words Approach for Multimedia Event Classification. INTERSPEECH.
49. https://en.wikipedia.org/wiki/Tf-idf
50. https://www.lexigram.io/lexipedia/medical-taxonomy-terminology-hierarchy/
51. https://www.nlm.nih.gov/research/umls/
52. https://searchhealthit.techtarget.com/definition/SNOMED-CT
53. Leaman, R., Islamaj Dogan, R., & Lu, Z. (2013). DNorm: disease name normalization with pairwise learning to rank. Bioinformatics, 29(22), 2909-2917.
54. Leaman, R., Khare, R., & Lu, Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. Journal Of Biomedical Informatics, 57, 28-37.
55. https://www.meddra.org/how-to-use/support-documentation/english
56. Ramos, Juan. (2003). Using TF-IDF to determine word relevance in document queries.
57. https://en.wikipedia.org/wiki/Conditional_random_field
58. Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R., & Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. Journal Of The American Medical Informatics Association, 22(3), 671-681.

59. Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. (2015). Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2267–2273.
60. Lee, Ji & Dernoncourt, Franck. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. 515-520.
61. https://developer.twitter.com/en/enterprise
62. https://scikit-learn.org/stable/
63. https://www.nltk.org/
64. https://tartarus.org/martin/PorterStemmer/
65. https://nlp.stanford.edu/projects/glove/
66. https://keras.io/