

Deep Learning for extracting Adverse Drug Reactions
from text documents

Dissertation submitted to Jawaharlal Nehru University

In partial fulfillment of the requirements

For the award of the degree of

MASTER OF TECHNOLOGY

IN

STATISTICAL COMPUTING (DATA SCIENCE)

BY

GAURAV KUMAR

UNDER THE SUPERVISION OF

Dr. ADITI SHARAN



School of Computer Science and System Science

JAWAHARLAL NEHRU UNIVERSITY

NEW DELHI- 110067

June 2022



Jawaharlal Nehru University

जवाहरलाल नेहरू विश्वविद्यालय

School of Computer Science & System Science

Jawaharlal Nehru University

New Delhi, 110067 INDIA

DECLARATION

I hereby declare that the dissertation work entitled “**Deep Learning for extracting Adverse Drug Reactions from text documents**” in partial fulfillment for the requirements of the degree of “**Master of Technology in Statistical Computing (Data Science)**” and submitted to “School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India”, is the authentic record of my own work carried out during the time of Master of Technology under the supervision of Dr. Aditi Sharan. This dissertation comprises only my original work. The matter personified in the dissertation has not been submitted for the award of any other degree or diploma.

DATE: 29 July 2022

Enrolment No- 20/10/MT/021

M.Tech. (2020-22)

SC&SS, JNU

NEW DELHI

Gaurav Kumar.

GAURAV KUMAR



Jawaharlal Nehru University

जवाहरलाल नेहरू विश्वविद्यालय

School of Computer Science & System Science

Jawaharlal Nehru University

New Delhi, 110067 INDIA


CERTIFICATE

This is to certify that the dissertation entitled “**Deep Learning for extracting Adverse Drug Reactions from text documents**” is being submitted by **Mr. Gaurav kumar** to “School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi-110067, India” in the partial fulfilment of the requirements for the award of the degree of “Master of Technology in Statistical Computing”.

 29-07-2022

(SUPERVISOR)

Dr. ADITI SHARAN


01-08-2022

(DEAN)

Prof. T.V. VIJAY KUMAR

ACKNOWLEDGMENT

I'm grateful to God for all of the blessings and opportunities that have come my way, as well as the love and support from people around me.

First and foremost, I want to express my thanks to my supervisor, Dr. Aditi Sharan, for providing me with the wonderful opportunity to work in her lab, which I was happy to join for my “M.Tech Dissertation”. I felt really privileged and honored to work under her supervision and in the field of my interest. Her constant support, motivation and encouragement have always been an inspiration for me to do my work in the best possible way. I’m really indebted to him for his constant guidance and for being considerate and a great guide.

I would like to thank the School of Computer and Systems Sciences and our Dean Prof. T.V. Vijay Kumar for providing me with all the facilities during my dissertation work.

A very special thanks to my seniors Miss. Shikha Verma, Miss. Samridhi Dev and Mr. Hemraj Kumawat, they taught me about my work from the start when I first joined the lab, as I was new to this field. Their command over techniques and their knowledge & love for their work always inspired me to do better. I'd like to express my gratitude to all of the faculty members who taught us the fundamental course subjects in order to assist us prepare for careers in various fields of science. A very special thanks to my lab mate Mr. Govind Soni for helping me on every step of work. Lastly, I’m grateful for having a very supportive family and friends who have always been my backbone till this stage of my life. Their love and support have taught me to be a better person each and every day.

Gaurav kumar.

Gaurav kumar

Abstract

In the Medical domain, the use of Adverse Drug Reaction (ADR) has been intensively used by doctors to identify and treat ADRs or the side effects. The use of Text Mining applications like Named Entity Recognition (NER) helps in the identification and extraction of related entities from Biomedical Textual documents. During the phase of extraction of the entities nowadays it has become a real challenge where there exist multiple continuous and overlapping entities. The purpose and objective of the thesis is to use proper methodology for extraction of the discontinuous and overlapping entities. The purpose and objective of the thesis is to use proper methodology for extraction of the discontinuous and overlapping entities. The proposed work includes the Construction of a Graph data structure for overlapping and discontinuous entities. The Maximal clique graph algorithm is used in the Mac model of the proposed work which has helped in the resolution of discontinuous entities and overlapping entities. The work has been performed on the CADEC dataset which is the benchmark dataset consisting of patient reviews taken from “Ask-a-Patient” platform.

Table of Contents

Declaration.....	i
Certificate.....	ii
Acknowledgment.....	iii
Abstract.....	iv
Abbreviations.....	vii
List of Figures.....	viii
List of Tables.....	ix
Chapter 1- Introduction.....	1
1.1 Introduction.....	1
1.2 Discontinuous NER.....	2
1.3 Brief Description Of MAC model.....	3
1.4 Problem statement.....	5
1.5 Thesis organization.....	5
Chapter- 2. Background and Motivation.....	7
2.1 Background.....	7
2.1.1 ADR.....	7
2.1.2 Information Extraction.....	8
2.1.3 NER Description.....	10
2.2 Corpus.....	11
2.2.1 Corpus Background.....	11
2.2.2 Biomedical Ontology Used For Corpus.....	12
2.2.2.1 MedDRA.....	12
2.2.2.2 SNOMED CT.....	15
2.2.3 Annotation Strategy.....	17
2.2.4 Construction of Corpus.....	18
2.3 Machine Learning Background and Approach.....	18
2.3.1 Approaches for Named Entity Recognition.....	18
2.3.2 Background for Deep Learning.....	19
2.3.3 Embedding's for Text Representation.....	19

2.3.3.1 Word Embedding's.....	19
2.3.3.2 Character Embedding's.....	20
2.4 Motivation.....	20
Chapter- 3. Related Work.....	22
3.1 Literature Survey.....	27
3.2 Dataset Creation and integration for ADR.....	27
3.3 ADR Prediction.....	28
Chapter- 4. Proposed Work.....	30
4.1 Clique and Maximal Clique.....	30
4.2 Grid Tagging Scheme.....	34
4.2.1 Segments Extraction.....	35
4.2.2 Edge prediction.....	35
4.3 Decoding Workflow.....	37
4.3.1 Algorithm 1 Decoding Procedure.....	38
4.3.2 B-k Algorithm.....	39
4.3.3 Explanation Of the Algorithm With Example.....	40
Chapter- 5. Experiment and Results.....	44
5.1 Dataset.....	44
5.2 Data Preparation and Implementation.....	46
5.2.1 Parameter Tuning.....	51
5.3 Evolution Parameters.....	52
5.4 Main Result.....	53
5.4.1 Overall Results.....	54
5.4.2 Result Comparison for Continuous an discontinuous.....	55
5.4.3 Impact of overlay Structure.....	56
5.4.4 Impact of Interval and Span Length.....	59
Chapter- 6. Conclusion And Future Work.....	63
7. Reference.....	64

Abbreviations

The following abbreviations are used in this manuscript:

ADE	adverse drug event
ADR	adverse drug reaction
BiLSTM	bidirectional long short term memory
CNN	convolutional neural network
CRF	conditional random fields
CUI	controlled unique identifier from UMLS
DI	drug indications
ID	identifier
LSTM	long short term memory
NA	not available
NER	named entity recognition
NLP	natural language processing
PoS	part of speech
PV	pharmacovigilance
SSI	sign/Symptom/Illness
TA	Twitter annotated corpus
TP	TwMED PubMed corpus
TT	TwMED twitter corpus
TTR	type/token ratio
WD	withdrawal symptom

List of Figures

Figure 1 The overlapping of entities and discontinues entity	2
Figure 2 “An example of the extraction process.”	4
Figure 3 An example of the extraction process	4
Figure 4 Figure 4 Standart structure levels in MedDRA	13
Figure 5 This is working flow of MedDra	14
Figure 6 Example: Using SNOMED CT Relationship	17
Figure 7 How CADEC Data Set Formed	18
Figure 8 Graph for Clique.....	31
Figure 9 Graph 2 for understanding clique	31
Figure 10 Explanation For Maximal Clique.....	33
Figure 11 After first loop iteration.....	40
Figure 12 Once all iteration ended for the upper loop we will get the graph.....	41
Figure 13 Work Flow of B-K Algorithm	42
Figure 14 The final result produced by B-K algorithm	43
Figure 15 This figure shows the flow diagram that how the original data set went under the preprocessing and then model implementation	46
Figure 16 these are the original folders in the CADEC corpus.....	47
Figure 17 This is the tokenized data of original files data	48
Figure 18 ann file formed by adding all individuals file in original file.....	48
Figure 19 This inline file is formed by merging the original folder files and text folder files	49
Figure 20 the splitting of the data set inline	49
Figure 21 This is the inside dictionary form of json file	50
Figure 22 These files created when we perform some embeddings and preprocessed	50
Figure 23 the last preprocess json file ready to inside the model	51
Figure 24 he comparison between TransE and Mac model Performance on different overlapping patterns	59
Figure 25 This figure shows the comparison between TransE and Mac model Performance on different interval length patterns	62
Figure 26 comparison between TransE and Mac model Performance on different span length patterns	62

List of Tables

Table 1 Overview of the related works in chronological order	35
Table 2 Overview of the related works in chronological order.	36
Table 3 A tagging example for segment extraction	44
Table 4 A tagging example for edge prediction	45
Table 5 The overall structure of the Mac model	46
Table 6 Corpus entities	56
Table 7 Confusion Matrix	61
Table 8 Statistics of datasets. S, M, and D respectively represent the number of sentences, total mentions, and discontinuous mentions. P denotes the percentage of discontinuous mentions in total mentions.	62
Table 9 Main results on three benchmark datasets. Bold marks highest number among all models	62
Table 10 Results on discontinuous entity mentions. In the Table, two scores are reported and separated by a slash (“/”). The former is the score on sentences with at least one discontinuous entity mentioned. The latter is the score only considering discontinuous entity mentions	63
Table 11	64
Table 12	64
Table 13 Statistics of interval length	67
Table 14 Statistics of span length	67

Chapter-1: Introduction

1.1 Introduction

In the medical sector, it has been observed that a lot of causes of mortality and morbidity are because of “Adverse drug reactions (ADRs).” Hence detection of ADR is a crucial concern in pharmaceutical safety. Adverse drug reactions are defined as the unwanted changes reflected in the body due to intervention of any drug. A rich source for ADR information is available in journal articles, social media, and drug reviews. However it is difficult to get the required relevant information specific to the needs of the user as the information is hidden within the text documents. Hence, ADR extraction is one of the recent topics in the text mining field in the health domain. Extracted ADRs can help doctors use drugs more rationally and reduce patient harm if identified early.

There are several techniques for extracting ADRs from text documents that include the traditional machine learning approaches, deep learning approaches, and natural language approaches. Extracting adverse drug reaction can be considered as a sort of “Named Entity Recognition” task.

In standard “Named Entity Recognition” problems, entities are considered to be continuous. Thus most of the NER approaches are focused on finding the boundary of the entity that encloses the entity. However in ADR entities are observed to be discontinuous. Detecting discontinuous entities is a challenging task and very limited work has been done in this area. This thesis focuses on identifying ADR indications, which include continuous and discontinuous entity.

Consider a post posted on askapatient.com “I still have pain in my arms and legs with much stiffness.” The objective is to collect the Adverse Drug Reaction mentions, namely, “pain in arms,” “pain in legs,” and “stiffness,” where “pain in arms” and “stiffness” are continuous ADR statement composed of continuous words and “pain in legs” is a discontinuous Adverse Drug Reaction mention composed of discontinuous words [2].

Researchers have looked into NER approaches in depth and proposed several State of Art (SOTA models) that are useful. The majority of previous techniques framed this problem as a sequence tagging problem, in which each token is given a label that describes its entity type. Their basic idea

is that an entity mentioned should be a brief text segment that does not overlap with other entities[3]. While this assumption is correct in most circumstances, this is not always the case, particularly in clinical corpora. Let have a brief look about the continuous and discontinuous entities. In the next section 1.2 we will see the discontinuous entities extraction from sentences.

1.2 Discontinuous NER

The task of Named Entity Recognition is to recognize and mine the entities from the textual data. One initial assumption that is typically made for continuous entities here is that the terms in the entities should come continuously in the text, also mostly it is assumed that these term should not overlap and are not nested. However there are many entities in the biomedical domain and specifically in ADR that do not follow these assumptions.

Consider an example referring to figure 1 . Figure 1 indicates two separate entity references E1() and E2() that are are discontinuous as well as overlapping with one another . Such entities are discontinuous

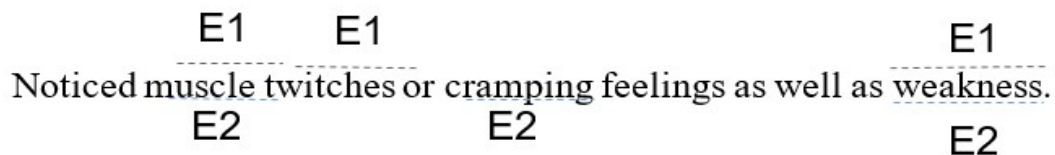


Figure 1 The overlapping of entities and discontinues entity

There are two types of SOTA models for finding discontinuous and overlapping entities. “Combination-based and transition-based” models identify all overlapping data first then learn how to mix these portions employing an independent classifier. “Transition-based” models tag data gradually the irregular extents via a series of Actions that shift-reduce [2]. Although these techniques have shown to be effective. There are still many issues : eg exposure bias[4]. Combination-based approaches, in particular, rely on the gold segments to direct the classifier during the classification during the training phase, the input segments inferred are provided by a trained model, resulting in the difference between training and inference. With transition-based

models, the current action builds on the previous golden actions during training, while the model builds the entire sequence of actions during testing. A skewed prediction will thus result in even greater discrepancies between the subsequent behaviors. Such a large disparity in performance might be detrimental.

Let us see the description of Mac Model, how its work and its use of maximal algorithm to resolve the problem of overlapping in the discontinuous entities.

1.3 Brief description of Mac Model

We present Mac, a discontinuous NER model based on maximal clique finding. The basic idea of the Mac model is that all the possibly discontinuous features mentioned in a sentence can logically construct a graph of segments by understanding the included continuous segments of them as nodes and connect segments with the same characteristics as edges. The “discontinuous NER” challenge is thus equal to identifying the graph's maximum cliques, which is an important graph theory problem. The remaining challenge is how to create such a segment graph.

In Mac, we split segment graph in two separate tasks: “segment extraction (SE)” and “edge prediction (EP)”. For segment extraction and edge prediction from an n -token phrase, two “ $n * n$ ” tag tables are generally produced, with each item representing the relations between two distinct tokens. Segment Extraction is thus considered a tagging issue in which tags are assigned to differentiate the border tokens of each segment, which aids in the detection of overlying segments. The challenge of line up the border tokens of segments contained in the similar entity is transformed to Edge Prediction. Overall, the label tables for Segment Extraction and Edge Prediction are produced individually and will be taken jointly by a maximum clique searching method to retrieve desired entity, constructing them immune to revelation bias [5].

Figure 2 depicts extraction process of MAC model through an example.

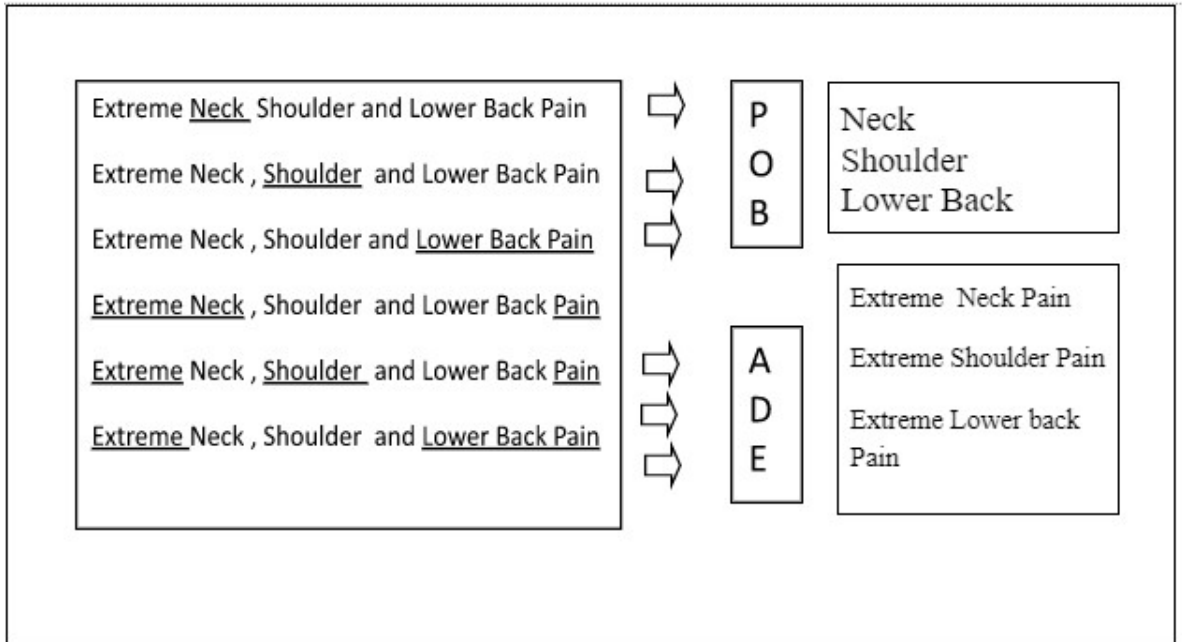


Figure 2 “An example of the extraction process.”

Figure 3 shows how reviews are taken and by using the BSI tagging scheme, it is tagged by parts of the body (POB) and Adverse Drug Event (ADE). From the figure 3 one can also see the segments for ADR named entity by joining the parts of tagged entity.* try to present figure properly

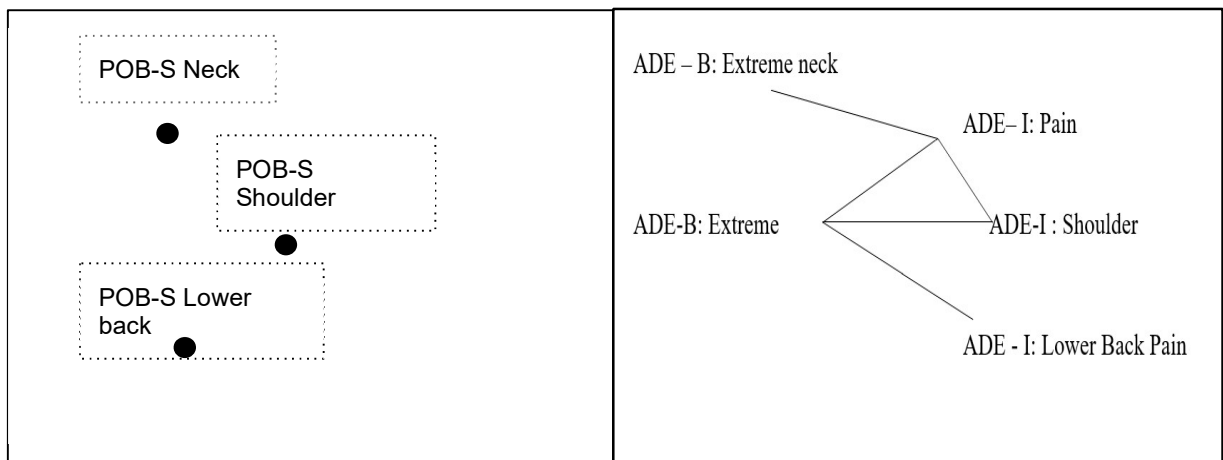


Figure 3 An example of the extraction process

There is a brief demonstration of this paper in diagrams 1 and 2 that shows how this paper works. As given example in figure 1 “Extreme neck ,shoulder and lower back pain” which is a review

comment of a patient on “askapatient.com” from that review comment we are using the BSI tagging scheme for tagging ADE (Adverse Drug Event) and POB (Parts of Body). Once we tag the reviews then we go for the second diagram where we use natural language processing techniques like pre-trained BERT model, embedding’s and maximal clique techniques for extracting discontinuous entities and finding overlapping.

1.4 Problem Statement

The objective of this problem is to recognize discontinuous and continuous NER entities for the adverse drug reaction problem. Most of the standard approaches treat entities as continuous segments, however in ADR a lot of entities extent discontinuous segments, also these segments can be overlapping. Identifying discontinuous and overlapping entities is still a research challenge. In our thesis we used a “deep learning” method using YelpBET model and developed MAC model for identifying such entities.

We have developed a model which will be able to effectively transition- based on a model that will identify discontinuous words without affecting original accuracy of continuous mentions. For solving to overlap we implement the concept of maximal cliques, in a segment graph. We proposed such a model whose task is to discover maximal cliques in a segment graph from the discontinuous NER. And we will verify this on CADEC DATA SET.

1.5 Thesis organization

This dissertation is divided into six chapters. Whereas in the first Chapter, It contains a brief introduction of our work about extraction of continuous and discontinuous entity findings and techniques we are going to use in it with some examples. And we will see an overview of the thesis and the importance of the proposed research plan for finding continuous and discontinuous findings. In this chapter we will also see about the briefs of all chapters in this thesis.

In chapter 2, It contains the Background and motivation of ADR and entity extraction for discontinuous entities which contain brief information about Adverse Drug Reaction, Information extraction, Name entity extraction, continuous and discontinuous NER and about the corpus.

Chapter 3 contains a brief survey of related study and literature pertaining to the area of the study of ADR, NER and continuous and discontinuous entity extraction.

Chapter 4 is the most important chapter of this thesis , it gives details about the methodology which has further 6 sections which give information about the tagging strategy, Grid tagging scheme, Segment extraction, Edge prediction, Algorithm of working and the implementation.

Whereas in the next chapter 5 is about the experiment which is also divided into 4 sections which give details about the Data set implementation which explain here dataset in mode into the formats for model. In the next section we discuss the results of the model and compare our model results to the TransE model. Then we go for the next section where we give details about the performance Analysis of the model as this model is good for discontinuous as well as continuous entity extraction so in the next section we see the impact of interval and span lengths.

In the last chapter 6 we will see the conclusion and offer future suggestions about the discontinuous and continuous entity and ADR.

Chapter:-2: Background and Motivation

In this chapter we are going to explain some terminologies related to our thesis like background and motivation of ADR and entity extraction for discontinuous entities which contain brief information about Adverse Drug Reaction, Information extraction, Name entity extraction and about the corpus then some machine learning approaches used in the biomedical domain.

Section 2.1 on the Background presents basics on the Adverse drug reactions and information extraction. While in ADR, It says what ADR is, how it is important and benefits in the medical domain. And information extractions contains presents the purpose of IE and how it is useful in name entity extraction. Section 2.2 discusses the corpus background, the corpus overview and biomedical ontology used for corpus then the annotation mechanism of corpus which was linked with somedCT and MedDRA. Further we discuss construction of the corpus. In section 2.3 we present an overview of the different machine learning approaches and techniques used in our work

2.1 Background

In the biomedical domain, “Named Entity recognition (NER)” is an significant task to mine the info from the text to support transactional research. A named entity can be utilized in many applications in “Natural language processing” some of them are:

- NER is extensively used in the biomedical domain such as DNA identification, gene identification, drug name identification, disease name identification, etc.
- Named entity recognition can be combined in the information retrieval model to optimize the retrieval process and the question-answering model to extract the relevant information.
- Named entity recognition is extensively used in social media domains such as opinion mining, text summarization, and finding the most relevant information. Hence we are going to explain some terminologies related to our thesis then some machine learning approaches used in the biomedical domain. Let us see a brief description of chapter 2 in the next section.

2.1.1 ADR (Adverse drug reactions)

In the medical sector, it has been observed that a lot of causes of mortality and morbidity is because of adverse drug reactions. Knowing which pharmacological targets are linked to ADRs can aid in the development of safer medications. Adverse drug responses (ADRs) detection is a crucial concern in pharmaceutical safety. ADRs endanger people's health, and the medical system and society incur huge financial losses as a result. ADRs can help doctors use drugs more rationally and

reduce patient harm if they are identified early. Numerous studies have looked into possible ADRs based on social media because of the resource's openness and timeliness. A major public health concern is adverse responses to commercially available drugs. [1] We looked at various research publications on ADR in order to improve patient care and safety, as well as contribute to the evaluation of benefits and harms, influence and risk of medicines, to encourage education and clinical training, as well as logical and safe medication usage. ADR is one of the decent topics in text mining field, a lot of digital data is now available in journals, social media and reviews about ADR, but all people are not able to directly access it because these data are sometimes in hidden form and some are in unpicked form. So this kind of information extraction can be done by text mining which can be helpful in the health domain. Text mining usually begins with a series of contiguous subtasks used to format text for the statistical analysis or pattern recognition phase. Subtasks consist of a basic set of low-level syntax tasks and a set of high-level tasks that include semantic processing based on the low-level tasks. Text mining has evolved into a tool for discovery, analysis, research, and management because it provides a mechanism for transforming free text into computable knowledge. Management of unused Pharmacovigilance can progress _pharmacovigilance, including the objective of identifying side effects of the drug.

2.1.2 Information Extraction

The purpose of information extraction is to generate a structured representation from unstructured text. There are several frequently explored IE tasks, including entity recognition, relation extraction, Coreference resolution, and event extraction. Entity recognition involves identifying and classifying noun phrases in text, based on a predefined set of categories (e.g. identifying and classifying person names, organizations, and locations). Relationship extraction involves object detection and classification (similar to object recognition) and determining semantic relationships between identified objects (e.g. determining if a city is located in a specific country, or determining the relationship between two people). Coreference resolution is the task of identifying all mentions of the same entity in a text (e.g. determining the proper noun referred to by pronouns). Event extraction involves identifying the phrase that indicates an event is present (called the "trigger"), classifying the trigger span, identifying argument (attribute) spans that characterize the event, and classifying the roles (relations) of the arguments (e.g. identifying the phrase

"outbreak" in a news feed as an indicator of a type of event) . Relation extraction, Coreference resolution, and event extraction tasks all involve identifying spans of interest and predicting links between identity spans, and there are similarities in the extraction architectures applied to these tasks.

This information extraction task is relevant to several clinical problems in IE. As an example, identification of protected medical information (de-identification of medical records) may be considered an object recognition operation [7]. Identifying medical problems, tests, and treatments and determining the relationship between these identified entities can be framed as a relation extraction task [8]. Identifying prescription drugs and associated adverse outcomes can also be explored as a relation extraction task [9], [10]. Characterization of many aspects of alcohol, drug and tobacco use, such as state, type, degree, and temporal information, can consist of event extraction problems [11]. This work approaches continuous and intermittent drug side effects and information extraction in the form of object recognition and event extraction tasks. There is a long history of IE in the general and clinical fields, and the method has evolved over time, starting with rule-based systems, moving to data-driven discrete modeling approaches, and now using neural networks. The approaches used in clinical IE tend to lag the methodologies used in the general domain. In a literature survey, Wang et al. [12] found that more than 60% of clinical IE studies from 2009-2016 used only rule-based systems. In contrast, Chiticariu et al. [13] found that rule-based systems represented less than 4% of recent general domain IE works from 2003-2012. In a survey of general domain conference papers, Young et al. [11] found that approximately 30%-40% of papers in 2012 used neural networks and that this proportion grew to approximately 70% by 2017. As is well used, neural approaches are becoming more prominent in clinical practice [14]. The subsequent section describes prominent neural modeling approaches, which are relevant to this work. Here in this work to extract information from the corpus, we are using a natural language processing technique known as Name Entity extraction. We are doing this also for discontinuous adverse drug entity extraction here. Let's have a look at what is NER and how it will work in this model.

Named Entity Extraction is one of the tasks related to Information Extraction. In coming section we discuss about NER in context of biomedical domain and ADR domain

2.1.3 NER Description

In this work in order to extract information from the drug review corpus, we are focusing on Name Entity extraction for ADR. We are doing this for continuous and discontinuous adverse drug entity extraction here.

The step of info extraction known as “Named Entity Recognition (NER)”, which focuses on tracing and categorizing named entities in text into pre-defined classes, is typically an unavoidable stage that is also required for some other “Natural Language Processing” (NLP) tasks. NER is concerned with detecting and categorizing named entities in text. Finding the intended entities in the document is often the phase that comes before relation cataloging while undertaking relation extraction tasks, for instance, which is another subfield of natural NLP. In addition to the general mining of entities such as names of people, places, and organizations [15], NER has been extensively researched in the field of biomedicine. Some examples of this research include the identification of biological composites [16], ‘genes and proteins’ [17], disorders [18], diseases [19], drugs [20], and adverse drug reactions . Utilizing the “Bidirectional Long Short-Term Memory (BiLSTM)” approach [10–12], or the combined Bidirectional “Long Short-Term Memory” and “Bidirectional Gated Recurrent Unit (BiGRU)” method , significant NER technologies have been created in order to detect Adverse Drug Reaction entities from social media. It's possible that the widespread interest in ADR detection from social media is due, in part, to the fact that it's easily accessible and updated in real time; yet, the material that's found in social media is also scant and unstructured [21–22]. There is a need for the development of NER methods in order to extract ADR-related elements from the free texts of ADERs which have a high information density. Named Entity Recognition (NER) is important in biological NLP. In pharmacovigilance, it's utilized to discover adverse drug occurrences in online customer reviews, notifying medicine producers, regulators, and physicians. NER can mine and encapsulate significant information from electronic medical data, such as formless doctor's notes. These apps need complicated references not found in general domains (Dai, 2018). Extensively used sequence tagging approaches incorporate two expectations that aren't necessarily true: (1) mentions do not nested or overlapped, thus each token can only be in the right place to one; and (2)

indications are continuous series of tokens. Nested entities identification resolves initial assumption breaches.

“Named entity recognition (NER) is the task of tagging entities in text with their corresponding type”, it is a sub-task of information extraction. In named entity recognition, many notations are defined, and commonly used notation is BIO notation. In BIO notation, B denotes the beginning of the entities; I denote the inside of the entities and O represents others, the non-entity tokens. Typically, BIO notation is used to differentiate the beginning and inside of the entity. Some examples are:

“ [(‘angiotensin-converting’, ‘B-GENE_OR_GENOME’), (‘enzyme’, ‘I-GENE_OR_GENOME’), (‘2’, ‘I-GENE_OR_GENOME’), (‘ace2’, ‘B-GENE_OR_GENOME’), (‘as’, ‘Other’), (‘a’, ‘Other’), (‘sars-cov-2’, ‘B-CORONAVIRUS’), (‘receptor’, ‘B-CHEMICAL’), (‘molecular’, ‘Other’), (‘mechanisms’, ‘Other’), (‘and’, ‘Other’), (‘potential’, ‘Other’), (‘therapeutic’, ‘Other’), (‘target’, ‘Other’), (‘has’, ‘Other’), (‘been’, ‘Other’), (‘sequenced’, ‘Other’)]

2.2 CORPUS

2.2.1 CORPUS BACKGROUND

The CADEC “CSIRO Adverse Drug Event Corpus” is a comprehensive annotated corpus of patient-reported “Adverse Drug Events”. This corpus is important for research on extracting information from social media, or more specifically data mining, in order to detect probable adverse medication responses from direct patient reports. [1]

This section describes the various aspects of datasets in the CADEC database like what CADEC database is, how the database was created and what are the entities in it.

For deeply understanding adverse drug reaction CADEC corpus has been explored.

The data has been mainly collected from AskaPatient website, which belongs to user reviews on medicines.

Patients can rate a medicine by filling out a thorough form based on the drug's brand name, such as Tamiflu. Consumer posts about the following 12 medications were supplied by AskaPatient: Flector Cataflam, Solaraze, Diclofenac Potassium, Pennsaid, “Voltaren-XR”, “Arthrotec”, “Pennsaid”,

“Flector”, “Cambia”, “Diclofenac Sodium Zipsor”, “Voltaren”, and Lipitor. [1] Dataset includes Drug, adverse effect, disease, symptom collected between 17-08-2001 to 17/10/2013. [1]

The majority of the information was gathered from the AskaPatient website, which is devoted to patient reviews of drugs. By completing a thorough evaluation form on a particular drug based on its brand name, such as Tamiflu, patients can score the treatment.

2.2.2 Biomedical ontology used for corpus

2.2.2.1 MedDRA

“MedDRA is a clinically-validated worldwide scientific terminology used by regulatory government and the regulated biopharmaceutical enterprise. The terminology is used through the complete regulatory technique, from pre-marketing to put up-advertising, and for facts entry, retrieval, evaluation, and presentation.”[34]

“Med=Medical

D=Dictionary for

R=Regulatory

A=Activities”

Why MedDRA is used ?

1. Facilitate the trade of scientific information via standardization.
2. Vital device for product assessment, tracking, conversation, digital statistics change, and oversight.
3. Statistics access and recovery and evaluation of medical knowledge approximately human medical products which include prescription drugs, biologics, vaccines, and drug-device mixture merchandise. [34]

Where it is used ?

“Regulatory Authority” and enterprise Records, person Case protection reports and protection Summaries, medical have a look at reports, Investigators ‘Brochures, center enterprise protection information, marketing packages, Prescribing statistics, and advertising guides.

MedDRA Structure

MedDRA Structure consisted of five levels as written below :

System Organ Class (SOC)

High Level Group Term (HLGT)

High Level Term (HLT)

Preferred Term (PT)

Lowest Level Term (LLT)

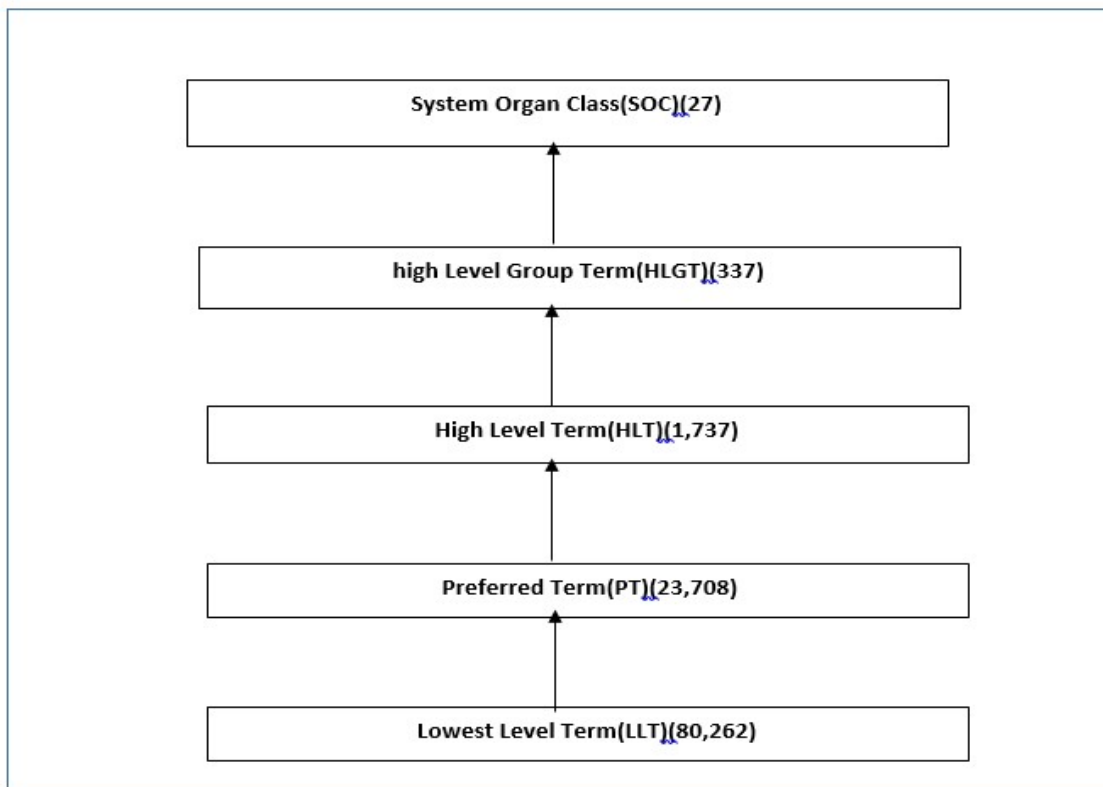


Figure 4 Standart structure levels in MedDRA

Figure 4 shows how each level is going to be specialized at every level as we see here the number of entities at LLT level is 80,262 while when it goes to preferred level it's become 23,708 and in last level System organ class it is only 27. We are giving an example in the next diagram to understand this.

In diagram 2 we showed an example of figure 1 how it works. As it is shown in diagram 2 that at the lowest level term (LLT) there are various kinds of Arrhythmia by generalizing them all to single Arrhythmia at preferred Term (PT). Then at high level term (HLT), this disease name changed based upon physiology, framework, pathology, etiology or function. Now High level terms are in turn linked to over High Level Group Term (HLGTs) to some common group. And this HLGTs passes this to the System Organ Group (SOC) which group them by etiology.

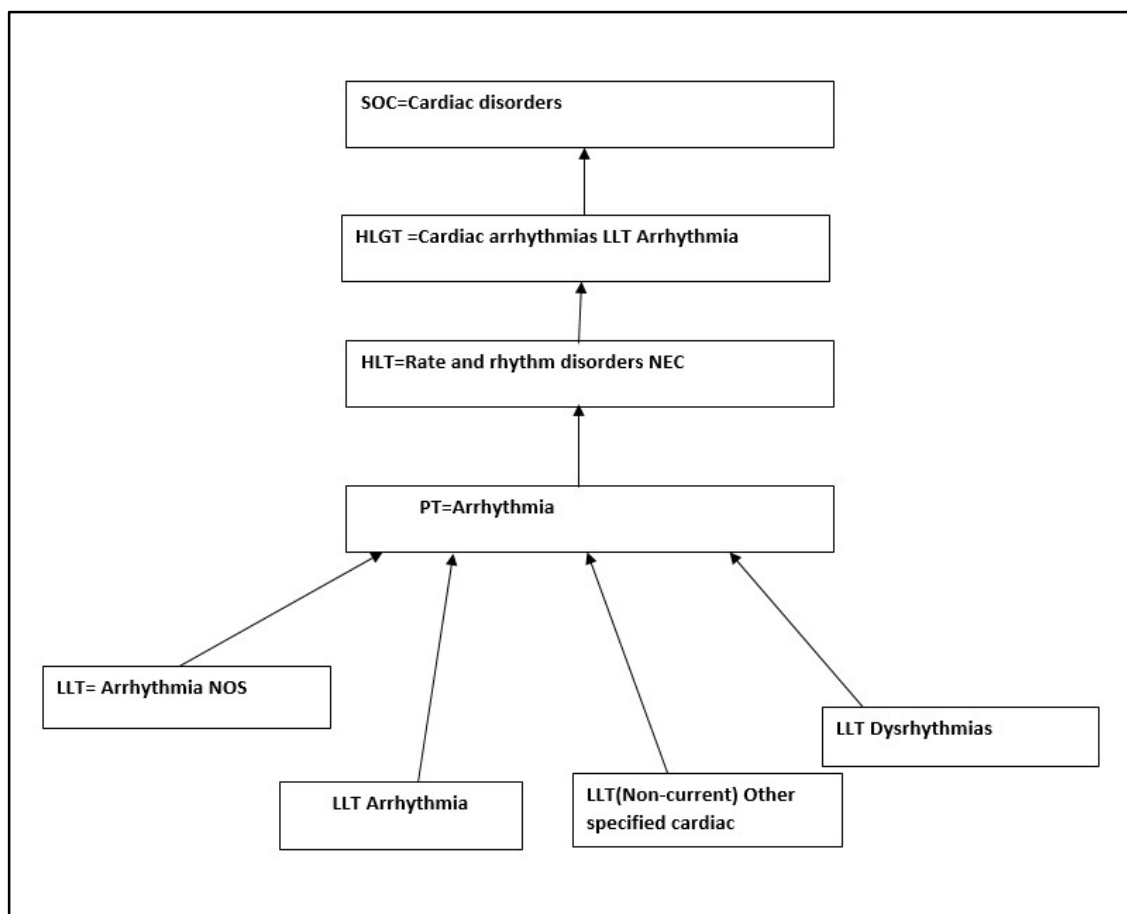


Figure 4 This is working flow of MedDra

2.2.2.2 SNOMED CT

SNOMED CT (Systematized Nomenclature of Medicine- Clinical Terms) is a clinical Terminology that provides codes, synonyms and definitions of clinical terms, and can be Accessed through the UMLS Met thesaurus.

It is an organized lists of a wide variety of clinical terminology defined with unique codes.

Perhaps the most comprehensive clinical terminology in the world.

SNOMED covers a wide no of medical terminologies for

- a. Disorders and finding (what was observed)

- b. Procedures (what was done)
- c. Event (what happened)
- d. Substance/Medication (what was consumed/administered)
- e. Pretty much anything that may be used to capture Medical data
- f. SNOMED is designed as an Ontology (Each Concept could have relationships with other Concepts).

Where will we be able to find SNOMED CT codes?

National Library of Medicine's UMLS is the one stop shop for SNOMED codes

SNOMED is now freely available for use for U.S. users

It is now maintained by International Health Terminology Standards Development

Organization (IHTSDO).

Concepts, Hierarchies & Relationships

- a.) A "Concept" is the basic unit in SNOMED Has a numeric representation(Concept ID),which is assigned arbitrarily Can represent anything that may have a possible use in recording clinical information The same Concept could have several "Synonyms" to accommodate variations in name E.g., "Myocardial infarction" could also be called "Infarction of heart" or just "Heart Attack"
- b.) All Concepts are divided in "Hierarchies" Hierarchies do not overlap Clinical Finding/Disorder, Procedure, Substance, etc. are all examples There are some 20+ main hierarchies, more can be added over time
- c.) "Relationships" between Concepts can be defined "Is a" is most common relationship other relationships could be defined as "Attributes" of a Concept

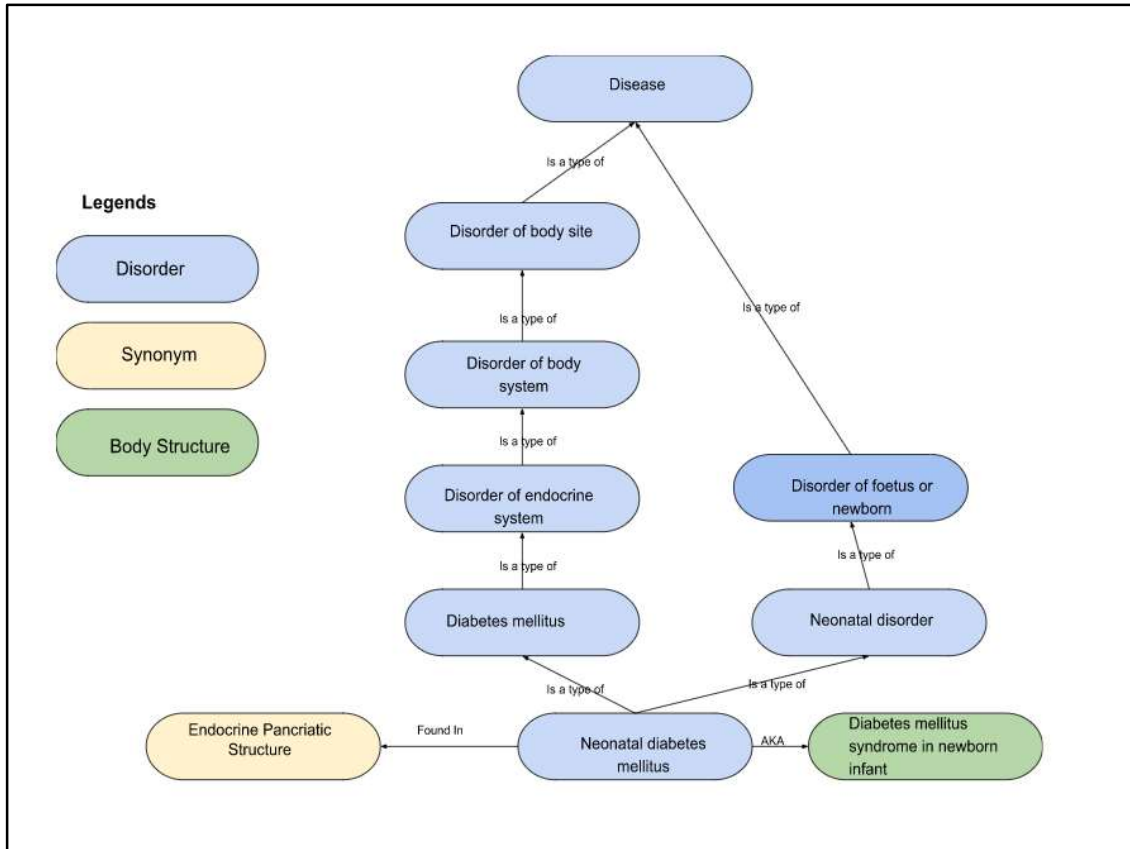


Figure 5 Example: Using SNOMED CT Relationship

2.2.3 Annotation Strategy

The data set is annotated mainly in dual steps:-

1) Entity Identification:-

In entity identification, it is identifying the mentions of entities of interest that is adverse reaction in the forums posts on the askapatient.com.

2) Terminology Association :-

At this stage of annotation it was linking the entities with the MedDRA and SNOMED CT.

2.2.4 Construction of corpus

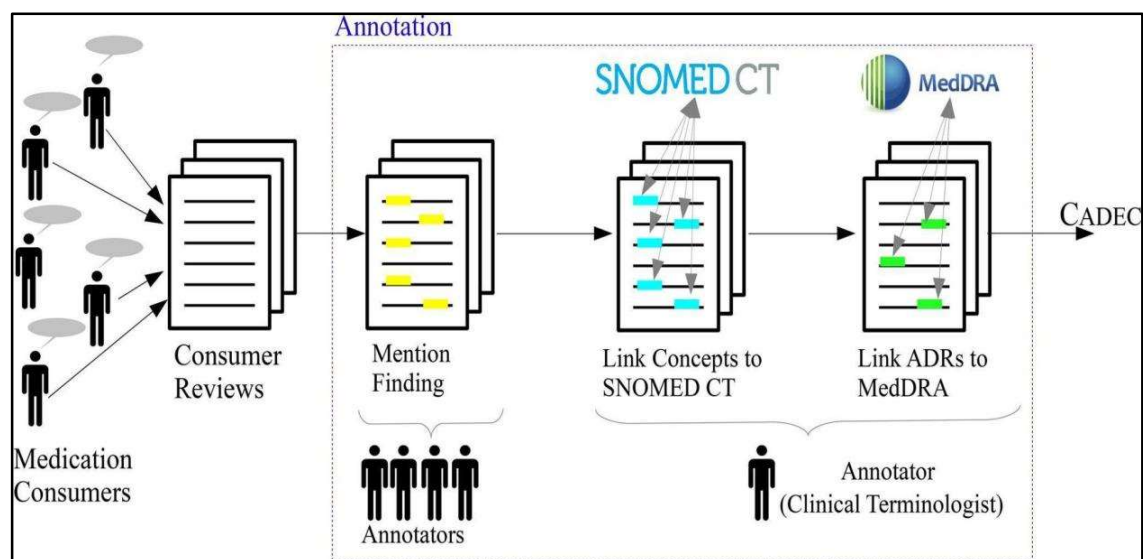


Figure 6 How CADEC Data Set Formed

The diagram 3 shows how the CADEC data set formed. At the very first step, medication consumers updated their reviews on askapatient.com where all the consumer reviews were stored. Then the manually annotators find the Mentions and make a corpus then this corpus was linked with SNOMED CT and MedDRA with clinical Terminologist

2.3 Machine learning Background and approach

2.3.1 Approaches for NER

For recognizing entities mainly “rule-based and machine”/”deep learning-based” methods have been proposed.

- Rule-based: Predefined set of rules are defined in the Rule-based technique. The rule-based Named Entity Recognition technique consists of two phases. In the first phase detecting and determining the entity, and the second phase selecting and extracting the entities from the text data. As rule-based techniques heavily rely on handcrafted tasks, it is a less efficient and time-consuming process.

- Machine learning/ Deep Learning-based technique: In machine learning-based, it tries to catch the pattern of occurrence of words in the data set. Machine learning-based models are heavily based on

the handcrafted feature, it is again a more time-consuming and challenging task to select appropriate features. Deep learning-based models, need not take care of the hand-crafted feature extraction. Many pieces of research show that sequential-based deep learning models like LSTM, BI-LSTM, etc. showed decent results in the Named Entity Recognition task.

2.3.2 Background for Deep Learning

Deep learning base: Neural network techniques are a fundamental requirement for automatic extracting the medical entity and “ADR” extraction whereas Name Entity Reorganization method helps a lot in extracting medical term from data set. There are some model proposed to identify accurately adverse drug reaction like “Long short-term memory (LSTM)” [5] is a kind of “recurrent neural network (RNN)” that uses adaptive gating to simulate interdependencies in sequential data and tackles the so-called vanishing or expanding gradients problem of vanilla “RNNs”[6]. Research has demonstrated that “bidirectional Long Short-Term Memory” (BiLSTM) and “Conditional Random Fields (CRF)” models [7] can reliably distinguish items in biological and clinical corpora. As a result, we investigated the utility of “BiLSTM-CRF” for detecting medical entities connected to ADEs in clinical narratives, also known as named entity recognition (NER). In deep learning for the extraction of ADR we use various word pre trained embedding’s. We use glove.2d pre trained word embedding’s in our model while the others kind of embed dings are “ELMO”, “word2vec” etc. For more understanding about the embedding we will see the section 2.3.3 where we discuss the embedding.

2.3.3 Embedding for text representation

2.3.2.1 Word Embedding

It maps the words to the small vectors called embedding. Embeddings are close to each other when words have a similar meaning or are related, far apart when they are not related. Word Embedding solves the sparsity problem. Once we have embedded our words into the small vector then it takes the same embedding to related words, it also reduces the dimensionality of text representation. “Embedding make it easier to do machine learning on large inputs like sparse vectors representing words. Ideally, an embedding capture some of the semantics of the input by placing semantically similar inputs close together in the embedding space. An embedding can be learned and reused across models.”

The idea behind the word embedding is to represent the text in numerical form. It captures the semantic relation amongst words. The pre-trained word embedding is learned from one task on a large dataset. Then after training, it is used for solving another similar task. These embedding allow transfer learning to be applied to text mining problems. These embedding are also able to detect the syntactic and semantic relation of words, which increases the performance of the model.

2.3.2.2 Character Embedding

Character Embedding is constructed at the character level in a language, in which we generate vectors for each character. Character level embedding can be thought of encoded lexical information and may be used to enhance or enrich word-level embedding. In character embedding, using a one-dimensional convolutional neural network we can find the embedding of words by their character level representation.

2.4 Motivation

“Named Entity Recognition” is one of the significant tasks in “Natural language processing” and has many practical applications. “Named Entity recognition” can be incorporated into many other tasks like information retrieval and mining to improve the results. In-text documents, every word is not equally important. Named entity recognition can help in categorizing the words according to their importance. The motivation of the research work are:

- Study the application of deep learning methods for NER in ADR domain.
- Discontinuous entity extraction
- Significance for ADR

The motivation behind this research work, Information retrieval, and extraction of discontinuous and continuous entries is a very interesting and challenging task, which becomes more challenging with the emergence of recognizing the Adverse Drug Reaction throwing new problems on a regular basis. It can be put to practical use in the biomedical domain. It will help biomedical practitioners, researchers, and students to extract relevant information from a vast corpus in a very efficient manner which consumes very little time also.

The others model generally undergoes from the problem of ambiguity due to narrow flexibility for an extended tag set as like in paper ([6]; [1];). As a development, Muis and Lu [3] use high graphs to signify business seasons and their combinations, but what has happened does not completely solve the problem of ambiguity[2]. Wang and Lu (2019)[4] introduced a pipeline framework that gets everyone who will run for office the scope of businesses and then combines them into organizations. By dividing the work into two dependencies steps, this method does not explain the problem of ambiguity, but at the instant it is interesting exposing bias. Recently, [2] built the sequence of the visual transformation action a non-continuous structure. By training time, predicting the basic truth in advance actions as a condition in the imaginary time he has selected exposure bias originates from basing the present course of action on the outcomes of earlier decisions. In this thesis, we first suggest one section on how to deal with a non-persistent NER when you are not tormented by a mysterious subject, he realized the consistency of training and interpretation.

Chapter -3: Related Work

3.1 Literature Survey

This chapter reviews the various methods that have been used to address ADR extraction in related tasks. At the same time, make a comparison and position Work in terms of relevance to the previous approach. Our goal is to explain the distinctive features, and above all, to adapt each task to our task. First, Table 1 and 2 summarizes the work-related in terms of the factors that appear to be the most differentiating to approach the ADR extraction. The elements defined in detail in the next section are listed below. Definition of

- i. AIR Extract: An entity involved in(indicated by P) where ADR is part of the text, reference to ADR as an entity (M), and related Alternative Dispute Resolution (R).
- ii. ADR classification method: conventional (T), deep learning (DL).
- iii. ADR characterization function: symbolic (S), dense (D).
- iv. Corpus (for ADR extraction): EHR (E), Social Media (SM), Scientific Publications (SP), Other (O) Text Genres and English (EN), Japanese (J), Swedish (SW) Language.
- v. Evaluation of ADR extraction: Holdout (HO) and k-validation cross-validation (CV) as evaluation schemes. F dimension (F)

Whenever possible, the positive class obtained from the holdout will be given as a rating measure. In other cases, the macro (M) or micro () value, or the area under the ROC curve (AUC) is also shown. The table always reports the results corresponding to the best-performing experiments.

Note, for example, that the definition of ADR itself is an impressive differentiator, but so are the characteristics of ADR, the approach used to extract them, or the evaluation method. In addition, Table 2.1 is for the sole purpose of summarizing excellent papers and may not cover all of the papers mentioned in this chapter. For example, the Table does not include works that did not use supervised machine learning. In contrast, it is interesting to work on ADR extraction (top of the table) as well as relationship extraction from closely related tasks (center) and related applied approaches within the medical field understood. Medical field (below).

				Corpora		Evaluation		
Authors	Definit ion	Classific ation	Charact erization	Textual genre	Lang uage	Sche me	Metric	Result
ADR extraction								
Aramaki et al.	R	T	S	E	J	10CV	F	59.8
Miura et al.	P	T	S	SP	J	5CV	F	37.5
Sohn et al. (2011)	R	T	S	E	EN	HO	F	74.5
Botsis et al. (2011)	R	T	S	SP	EN	HO	<i>F_M</i>	81.3
Gurulingappa et al. (2011)	P	T	S	SP	EN	10CV	F	76.0
Gurulingappa et al. (2012a)	R	T	S	SP	EN	HO	F	87.0
Karlsson et al. (2013)	P	T	S	E	SW	10CV	AUC	87.0
Patki et al. (2014)	P	T	S	SM	EN	10CV	F	65.2
Ginn et al. (2014)	P	T	S	SM	EN	10CV	F	76.6
Zhao et al. (2014)	P	T	S	E	SW	10CV	AUC	71.7
Zhao et al. (2015)	P	T	S	E	SW	10CV	AUC	76.3
Friedrich and Dalianis	P	T	S	E	SW	10CV	F	67.0

(2015)								
Li et al. (2015)	R	T	S	SP	EN	10CV	F	51.1
Sarker and Gonzalez (2015)	P	T	S	SP	EN	HO	F	81.2
Nikfarjam et al. (2015)	M	T	S,D	SM	EN	HO	F	82.1
Lin et al. (2015)	M	T	S,D	SM	EN	HO	F	62.5
Henriksson et al. (2015a)	R	T	S,D	E	SW	HO	F	27.2
Henriksson et al. (2015b)	P	T	D	E	SW	10CV	AUC	94.0
Zhang et al. (2016)	P	T	S,D	SM	EN	HO	F	54.9
Huynh et al. (2016)	P	DL	D	SP	EN	10CV	F	87.0
Stanovsky et al. (2017)	M	DL	D	SM	EN	HO	F	93.4
Lee et al. (2017)	P	DL	D	SM	EN	HO	F	64.5
Tutubalina and Nikolenko (2017)	M	DL	D	SM	EN	HO	FM	79.8
Akhtyamova et al. (2017)	P	DL	D	SM	EN	HO	F	54.2
Cocos et al.	M	DL	D	SM	EN	HO	F	75.5

(2017)								
Gupta et al. (2018)	M	DL	D	SM	EN	HO	F	75.1
Wunnava et al. (2018)	M	DL	D	E	EN	HO	F	63.5
Masino et al. (2018)	P	DL	D	SM	EN	HO	F	45.7
Fabregat et al. (2018)	R	DL	D	SP	EN	10CV	FM	75.6

Table 1 “Overview of the related works in chronological order”

				Corpora		Evaluation		
Authors	Def initi on	Clas sifi cati on	Cha ract eriz atio n	Tex tual genr e	Lan gua ge	Scheme	Met ric	Res ult
Relation extraction applied to other domains								
Celli (2010)	R	T	S	O	EN	10CV	F	26.7
Zeng et al. (2014)	R	DL	D	O	EN	HO	F_M	82.7
Ebrahimi and Dou (2015)	R	DL	D	O	EN	HO	F	82.7
Nguyen and Grishman (2015)	R	DL	D	O	EN	HO	F_M	82.8
Miwa and Bansal (2016)	R	DL	D	O	EN	HO	F_M	85.5
Zheng et al. (2016)	R	DL	D	O	EN	HO	F_M	83.8
Zhou et al. (2016)	R	DL	D	O	EN	HO	FM	84.0
Katiyar and Cardie (2017)	R	DL	D	O	EN	HO	F_μ	55.9
Christopoulou et al. (2018)	R	DL	D	O	EN	HO	F_μ	64.2
Ren et al. (2018)	R	DL	D	O	EN	HO	FM	87.4
Le et al. (2018)	R	DL	D	O	EN	HO	FM	86.3

Table 2 “Overview of the related works in chronological order.”

Summary of the correlated works in sequential order, separating those devoted to ADR extraction (in the top), those within the medical domain (in the middle) and those out of the medical domain (in the bottom). The different values are: Presence (denoted as P), Mention (M), Relation (R), Traditional (T), Deep Learning (DL), Symbolic (S), Dense (D), EHR (E), Social Media (SM), Scientific Publications (SP), Others (O), English (EN), Japanese (J), Swedish (SW), Hold-Out (HD), Cross-Validation (CV), “F-measure (F)”, Area Under the ROC Curve (AUC), together with macro (M),

micro (μ).

3.2 Dataset creation and integration for adverse drug reaction

Seminal contributions have been made by researchers in integrating information from different datasets. The relationship among genes, alleles, medications, adverse drug reactions, populations, diseases, and allele frequencies has been covered in a repository created by the authors. [24]-[26]. The problem of mechanically obtaining data on hereditary variants and their consequences for medication responses from PubMed précises is addressed by some researchers [27]. The global library includes correlations between allele frequencies, medications, diseases, side effects, populations, and genetic variations and variants. [25] Focused on Using Collaborative Filtering to Apply Similarity Reference to Integrate Manifold Evidence Sources to Predict “Adverse Drug Reactions”. [28] Suggested a method of creating side-effect profiles from drug-association table and integrating it with chemical and biological data. They combined and annotated information from openly available databases on medications, chemicals, protein targets, illnesses, side effects, and pathways to create a semantically connected network with over 290,000 nodes and 720,000 edges.[29], [30] Effective growth will lead to the development of competence and capability.[31]

Emphasized on selection and linkage of data sources relevant to pharmacogenomics. Semantic networks were also build by integrating and annotating data from public datasets relating to drug’s biological and physical properties, their side effects. Studies dealing with semantic networks emphasized on embedding, Knowledge graphs and graph embedding [26], [28], [29]. Random forest and graph kernel algorithm was used to identify and prioritize pharmacogenes that are valid. Various machine learning algorithm have been applied to predict ADR like K-nearest neighbor, SVM, decision tress, Logistic regression.

3.3 Adverse drug reaction prediction

The prediction algorithm aims to forecast new cases by using information about medications known to result in an ADR. According to the research paper under consideration, there are three categories for state-of-the-art techniques:

3.3.1. Knowledge graphs and graph networks

The knowledge graph (KG) encodes data from the available literature, and rough edges reflect clinical concepts and their relationships, respectively. Previous research has shown how beneficial such literature-derived KGs are for predicting ADE. To allow inference in large and complex KGs, the latest approach uses graph embedding. It encodes the essential properties of a particular graph globally into a vector representation of its vertices. With such a representation, the relationships between clinical concepts can be calculated algebraically using vector similarity. Additionally, for ADR prediction tasks, these descriptions can be employed directly as a function in machine learning models. The author [32] proposed a weighted version of the Deep Walk and Trans algorithms for feature learning of literature-driven knowledge graphs with three kinds of connections (“drug-drug, interactions drug-protein, protein-protein”). After feature learning, three classification models were used: ANN, linear regression, and random forest. [33] Knowledge graphs were created using openly available data containing information on drugs, their objective clinical suggestions, proteins, and known Adverse Drug Reactions. The edge among the drug node and the ADR node is missing for the unknown ADR of the drug. To distinguish between known causes of “ADR” and all other medicines in the graph, they utilized Fisher's exact test to determine which features were enriched. This serves as an input to a prediction algorithm that, by assuming the presence of any missing edges in the plot, forecasts an unknown ADR [31] application of a casual walk algorithm to a linkage of drug and “ADR” nodes to predict an unknown ADR. Here, the drug ADR edge represents a known ADR, and the drug-drug edge indicates drug target similarity, but new real-world ADRs have not been clinically validated data [24]

To characterize the information on medications, ADEs, and their relations, a bipartite network was built. In this network, nodes indicate drugs or ADEs, and edges indicate the association of known drugs with ADEs. They developed a “logistic regression model” to forecast previously unidentified drug-ADE correlations. [32] Proposed a unique method that combines deep learning and a biological tripartite network to predict drug-ADR connections.

By searching for compounds that are structurally related to already approved medications, similarity-based techniques can predict ADR [33], [34]. These techniques, while relatively simple to use, are ineffective if the suggested drug's structure differs from the structure of the existing drug. It does not optimize for each individual ADR and gives equal weight to all structural features.

Additionally, these models are more challenging to interpret in order to use machine learning to identify the chemical components that cause ADR.

The flexibility of machine learning algorithms to handle a variety of data types, including chemical and genetic data, pharmacological or phenotypic information, has led to an increase in interest in these methods in recent years for predicting side effects. A very fruitful field of machine learning research has been deep learning. “Machine learning” approaches can be alienated into two types: unsupervised and supervised learning. Typical “Supervised learning” methods include rule induction, neural networks, “Naïve Bayesian classification”, “support vector machines”, regression, decision trees, etc. The “Artificial Neural Network” is an example of supervised learning used by the researcher for “ADR prediction” considering it as a multiclass classification. [34] Have utilized random forests for predicting ADR occurrence. The machine learning-based methods make use of molecular fingerprints like the circular and PubChem fingerprints. Bresso and co. To anticipate and validate ADR profiles, we created a record of medication, Adverse Drug Reaction, and goal knowledge. We then employed “decision trees” and “logic programming” (rather than individual ADRs). Using FAER S3, ADRs. [29] used the structural characteristics of the “drug-ADR” relation, together with the biochemical and taxonomic characteristics of pharmaceuticals, as features to train a “logistic regression classifier” to guess unknown ADRs for commercially available drug. Created a neural fingerprint technique for ADR prediction within a concurrent deep learning system. They used the attention framework to analyze the deep learning framework and feature analysis to determine which organizations within the medication molecules are precisely linked to a given ADR. The major goals of the deep learning model were to predict and identify potential adverse drug reactions for new medications. have developed a “bi-LSTM and CRF-based” model for predicting ADR.

Chapter:-4 : Proposed Work

Our proposed work has been designed to extract ADR based entities from a drug review corpus. The extracted entities can be continuous, discontinuous or overlapping. Handling discontinuous and overlapped entities is a challenging research problem. As ADR contains several discontinuous and overlapping entities, our approach is very useful for NER for ADR domain.

Our work is based on Graph based approach. In this approaches terms in entities are represented in form of graph, edges representing the segments connecting the terms. Once the graph is constructed, deep learning method (BERT) is used to extract the entities. Further maximal clique approach is used to handle the overlaps between the entities.

This chapter is divided in three sections. As Graph cliques form an important part of the approach, section 4.1 describes clique, maxclique approach and how does it help in our proposed work. Section 4.2 presents the Grid tagging scheme used for tagging the corpus. Section3 presents the decoding workflow.

4.1 Clique and Maximal Clique

In this section we will discuss the terminologies which are used in this thesis like what is clique and maximal clique and how it helps in this algorithm.

Let's come to understand what clique, maximal clique and how its algorithm works. We will also see the Gridding Scheme Section in which we see Segment Extraction and "Edge Prediction". In Segment Extraction, we will see how we will assign tags to the entity like "extreme" as ADE-B means extreme is "adverse drug event begging" whereas in "Edge Prediction" we will build the link among Segments of the same entities. Then we will go to the decoding section where we define algorithms and explain the algorithm with an example to execute the above methodology and extract continuous and discontinuous entities.

"A **clique** is a subset of vertices of an undirected graph G such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete. Cliques are one of the basic concepts of graph theory and are used in many other mathematical problems and constructions on

graphs”. Let’s see an example, a set $C1$ is a clique of Graph $G1$ (Figure 8) iff $C1$ is subset of G & every pair of distinct vertices and adjacent in G as G is shown in figure 8 below.

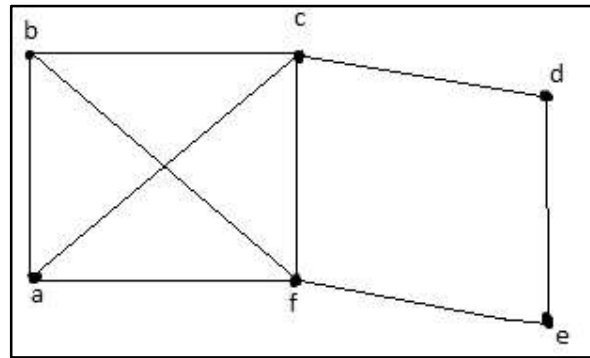


Figure 7 Graph for Clique

Graph- $G1$

$C1 = \{a, b, c, f\}$ Here the $C1$ is Clique it is because it is subgraph of $G1$ (Figure 8) and $ab, ac, af, bc, bf,$ and ba

These all are adjacent vertices.

Similarly $C2 = \{a, b, c\}$ is also Clique graph it is also satisfying condition of clique that it is subgraph and all the vertices are adjacent .

Now we come to the what is maximal clique ?

“A maximal clique is a clique that cannot be extended by including one more adjacent vertex, meaning it is not a subset of a larger clique. A maximum clique (i.e., clique of largest size in a given graph) is therefore always maximal, but the converse does not hold.”

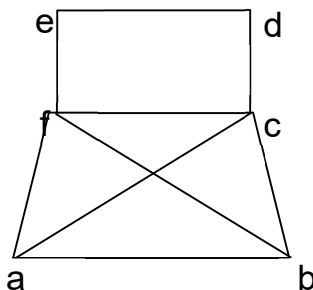


Figure 8 Graph 2 for understanding clique

$C3=\{a, b, f\}$ is a 3 vertices Clique as per we explained above now adding one vertices “c” to the $C3$ clique we get $C4=\{a, b, f, c\}$ which also a Clique so by adding new vertices in $C3$ giving new clique $C4$ so $C3$ is not a maximal clique and now we check $C4$ is maximal clique of the $G-2$ graph is or not. Let us add a new vertices “e” in the $C4$ clique we will get $C5=\{a, b, f, c, e\}$ as we check the conditions of clique which is not satisfy by clique $C5$ so we can say that $C4$ is maximal clique of Graph $G2$ (Figure 9) and in one graph there may be more than one maximal clique. We explain the above things in the figures 10 below.

This demonstrates that no further vertexes can be added that are similar to the relationship between the segments in a “discontinuous entity” since every vertex in the maximum clique has a close relationship with one another.

On the basis of above explanation realization, we indicate that “discontinuous NER” may be consistently considered as the process of identifying maximum cliques inside a segments graph, in which nodes denote segments that either constitute objects all on their own or appear as components of a whole entities without continuity, whose boundaries link individual segments that are part of the same entity mention.

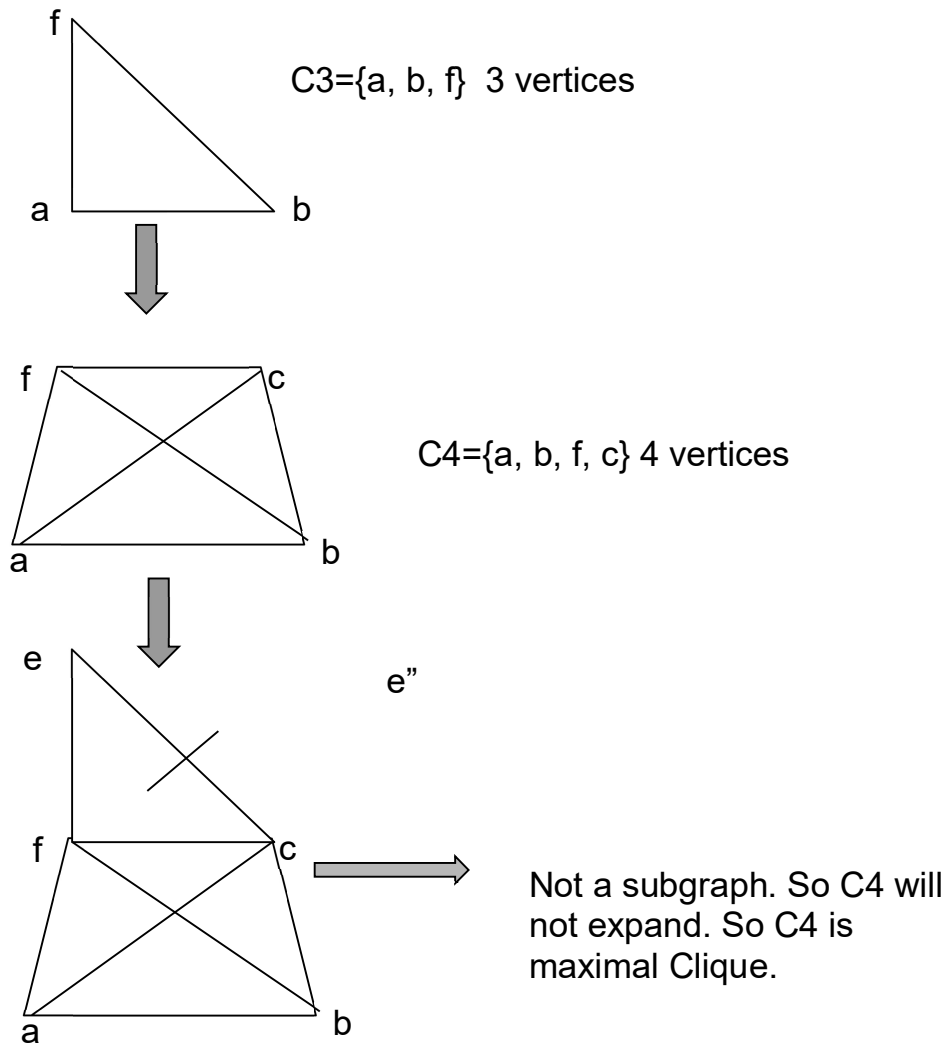


Figure 9 Explanation For Maximal Clique

In practice, discontinuous NER is broken down into two separate tasks known as segment mining and edge estimate. These jobs served to construct the segment graph's nodes and edges, respectively. Their predicted outcomes may be created individually using recommended grid tagging system, and they will be expended together in order to form a "segment graph". This will allow maximum clique discovery approach to be employed in order to recover the entities that are required. Figure 2 presents an summary of the mining process in its entirety. Onwards from here we will see the grid tagging scheme in which we will see segment extraction and Edge prediction then after grid tagging we will see its decoding workflow. After explaining the above term we will

see a “Maximal clique discovery” based discontinuous “NER model” based on the above tagging scheme. Now let’s have a look at Grid Tagging Scheme

4.2 Grid Tagging Scheme:

As reviewing paper, Wang et al[2] we inspired and based on that, an unusual grid tagging approach, we constructed a single-stage segment extraction and edge prediction system.

If an n-token sentence is provided, our method will construct “n * n” tag table by first listing all of the likely token pairs, and then appointing each token pair with a tag or tags based on how closely they are related to one another. This will allow us to categorize the tokens in the sentence (s). Take note that according to the predefined tag set, one token pair might contain many tags at the same time.

4.2.1 Segment Extraction

As we see in the paper acl (2020) and example one where entity mentions could overlap with each other which was a problem in previous paper how to get the solution of extraction overlap entity. To develop such a model which has the ability of mining such overlying segments, we develop a tag table with two dimensions: we have a table 3 is an example of such a tag table of 2-dimensions. A set of labels that will be assigned to a pair of tokens that will be denoted by (a_i, a_j) if a segment from a_i to a_j goes to the relevant groups. As a seeing “ $j > i$ ”, we will remove the lower triangle region of the label table because we cannot traverse back here. So $(n^2 + n)/2$ numbers of grids will be formed in real for the n number of token sentences.

In actual practice, The BIS (The Bureau of Indian Standards a tag set suggested that annotating the part of speech in Indian language should make use of a standard tag set) “tagging scheme” is used to describe that if a given segment is incessant entities then it is mention as (“X-S”) or detects at the beginning then present as (“X-B”) if it is inside the (“X-I”) of a discontinue entity of kind X.

Let us understand this with our example 2. The sentence is “Extreme neck, shoulder, and lower back pain”. Here lower back is allocated with the label “POB-S” (Parts of body - Beginning) since” lower back is a continuous entity of kind of body. Similarly “neck” is assigned as POB-S since it is also continuous.

And the tag of (Extreme, neck) is “ADE-B” as “Extreme neck” is the start of a segment of discontinuous mention of “Extreme neck pain” of kind of Drug event ADE.

Now, neck is considered as an entity since there is “POB-S” label in the place of (Neck, neck), Hence the segment overlying extract issues is resolved.

	EXTREM E	NECK	,	SHOULDER	AND	LOWER	BACK	PAIN
EXTREME	ADE-B	ADE-S						
NECK		POB-S						
,								
SHOULDE R				POB-S ADE-S				
AND								
LOWER							POB-S	ADE-I
BACK								
PAIN								ADE -I

Table 3. “A tagging example for segment extraction”

4.2.2 Edge Prediction

The process of constructing the relationships among segments of the similar entity by involving their margin tokens is considered here as edge prediction. Here we define tagging scheme in the same way as we define in above section we define the tagging scheme as follows:-

- 1) X-H2H denotes “head to head” which point to it location in a place (ai,aj) where ai and aj are respectively the starting of the tokens of two sections which is comprised of the similar entity of kind X.

2) “Tail to tail” denoted as “X-T2T” it means that it will look on the endings of the tokens.

Let’s see an example. The token “Extreme” has the “ADE-H2H” and “ADE-T2T” relation to the “shoulder” and “pain”. Since the kind of the “discontinuous entities” mentions “Extreme shoulder pain” is an “Adverse Drug Event”. Similarly we see the relations of “Extreme” to have the ADE-H2H relations with the “lower back”. Same way “Extreme” has ADE-H2H & ADE-T2T relation with token “Pain” with the same technique as the matrix shown in table 4.

	EXTREME	NECK	,	SHOULDER	AND	LOWER	BACK	PAIN
EXTREME				ADE-H2H ADE-T2T		ADE-H2H		ADE-H2H ADE-T2T
NECK								ADE T2T
,								
SHOULDER	ADE-H2H ADE-T2T							
AND								
LOWER	ADE-H2H							
BACK								
PAIN	ADE-H2H ADE-T2T	ADE-T2T		ADE-H2H ADE-T2T				

Table 4 “A tagging example for edge prediction”

4.3 Decoding workflow

The decoding technique may be reduced to its most basic form, which is described in Algorithm 1. The segment tagging table (denoted by the letter S) and the edge tagging table (denoted by the letter E) of a phrase (denoted by the letter T) make up the inputs, respectively. Table 5 shows the decoding flow of our algorithm.

To begin, we must first go through the steps of decoding S in order to get all of the typed segments. After that, we construct what we will refer to as the segment graph G. In this graph, segments that have been decoded to indicate that they belong to the same entity (represented by E) are connected through edges, as shown in the illustration in figure 2. In the same way, we are able to directly produce an ongoing entity mention from the clique of single-vertexes, and we are able to recover discontinuous entity mentions by concatenating sections in each multiple-vertex clique, following the order in which they appeared in T in their original sequential order. This allows us to produce a “continuous entity” mentioned directly from the single vertex clique. We will find the most cliques in G by using the time-consuming yet dependable B-K backtracking method (Bron and Kerbosch, 1973).

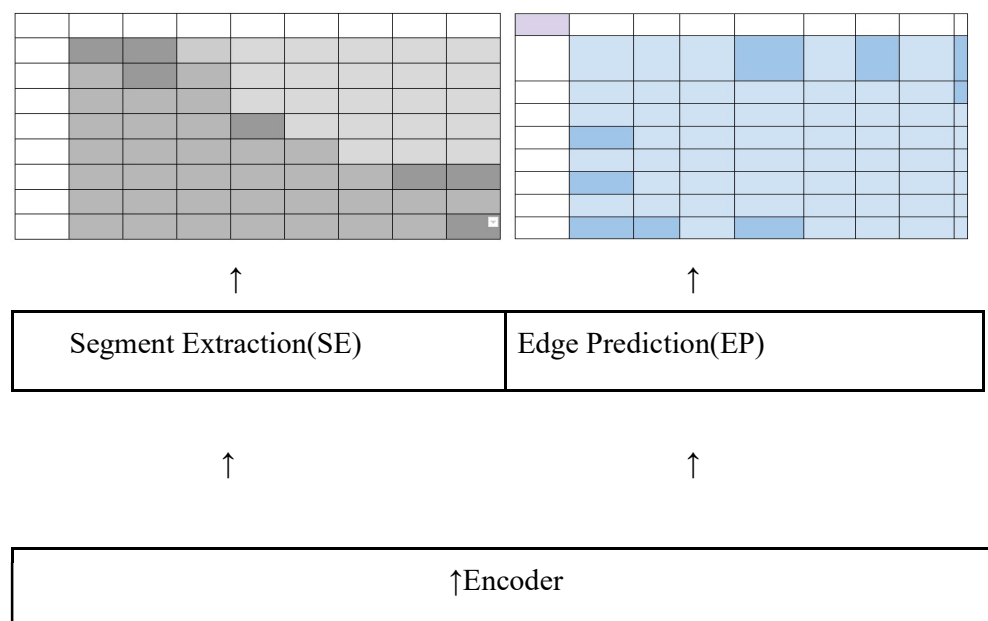


Table 5 “The overall structure of the Mac model”

4.3.2 Algorithm 2 B-K algorithm

The graph G is the input.

The whole collection of maximum clique C is the output.

- 1: Using?, initialise C as well as the two vertex sets R and X .
- 2: Define the node set, P , of the graph G .
- BRONKER, number 3, as a function (R, P, X)
- 4: if $P = ?$ & $X = ?$ then
5. Include R in C
- 6: end if 7: for $v \in P$ do
- 8: Define $N(v)$ as the set of all of v 's neighbours;
- 9: BRONKER($R \cup N(v), P \setminus N(v), X \cup N(v)$)
- 10: $P \setminus v$
- 11: $X \setminus v$
- 12: end for
- 13: end function
- 14: BRONKER(R, P, X); / This will call the BronKer function.
- 15: send in the C

4.3.3 Explanation of the algorithm with example

Let us understand the given algorithm with an example shown in the segment table 3 and edge table 4.

As per the Algorithm, we get initial value are as.

Edge $E = \{\phi\}$

Entity $R = \{\phi\}$

$N = \{ \text{"Extreme " , " Neck " , " , " , " Shoulder" , " and " , " lower" , " back" , Pain" } \}$

After the first iteration of upper for loop we get.

$E = \{ (\text{Extreme, Shoulder}), \}$
 $\quad (\text{Extreme, Lower back}),$
 $\quad (\text{Extreme, Pain}) \}$

So, this shows the edge between them the vertices.

Hence we get a figure 11 for the first upper loop iteration.

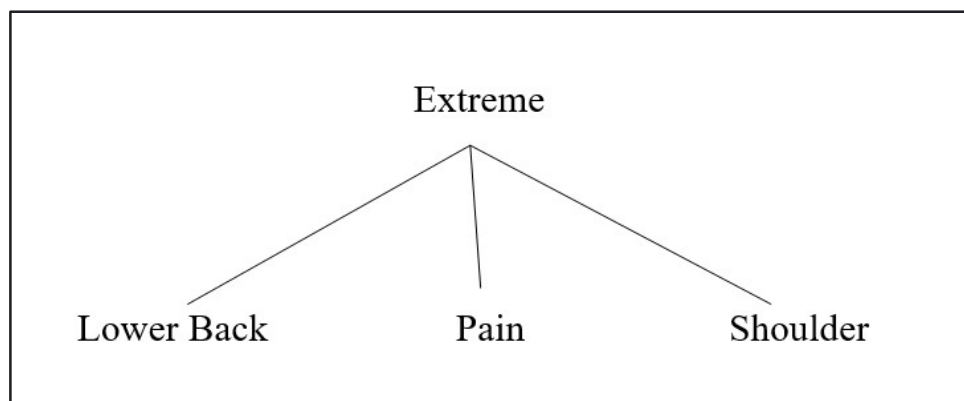


Figure 10 After first loop iteration

Once all iteration ended for the upper loop we will get the figure 12.

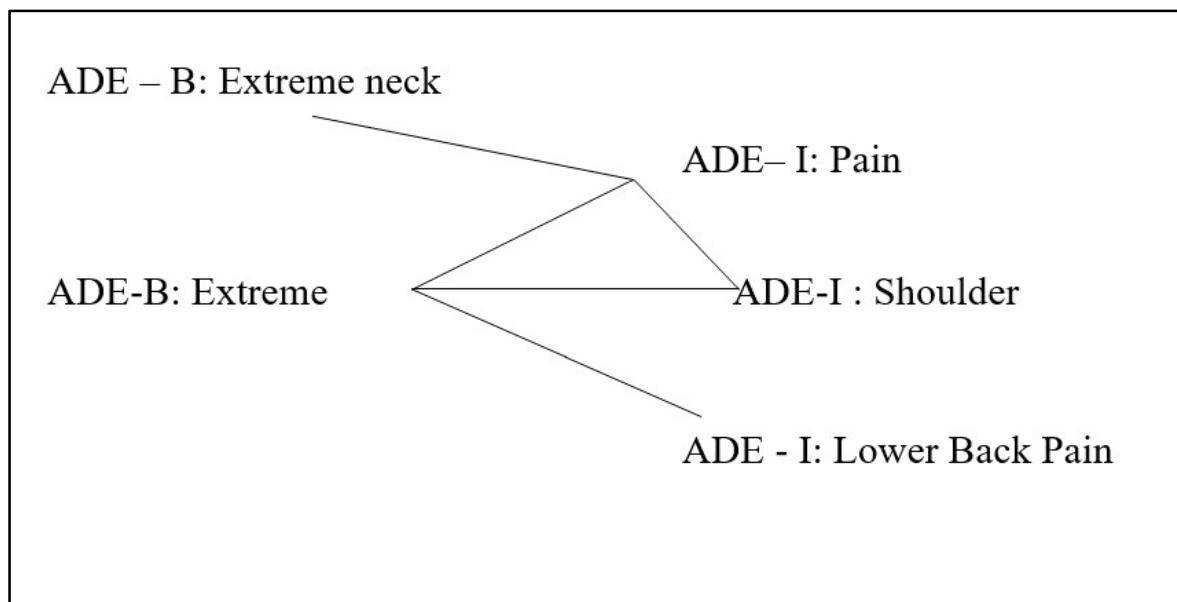


Figure 11 Once all iteration ended for the upper loop we will get the graph

Now we have reached step 12 of algorithm 1, we have to go 1 for algorithm 2. Where the upper graph is as input.

As per our convenience we give vertices names in numerical value to easily tack.

Extreme Neck as 1, Pain as 2, Extreme 3, Shoulder as 4, and lower back pain as 5.

Algorithm 2 is for the minimal clique

Now start the algorithm in 2 steps.

As per algorithm 2 we have initial value are as:-

$R = \{ \}$

$X = \{ \}$

$P = \{ 1,2,3,4,5 \}$

Now at step 3 a function which is called by step 14. As we see here it is a recursion function. So we will get this function by the figure 13 given below.

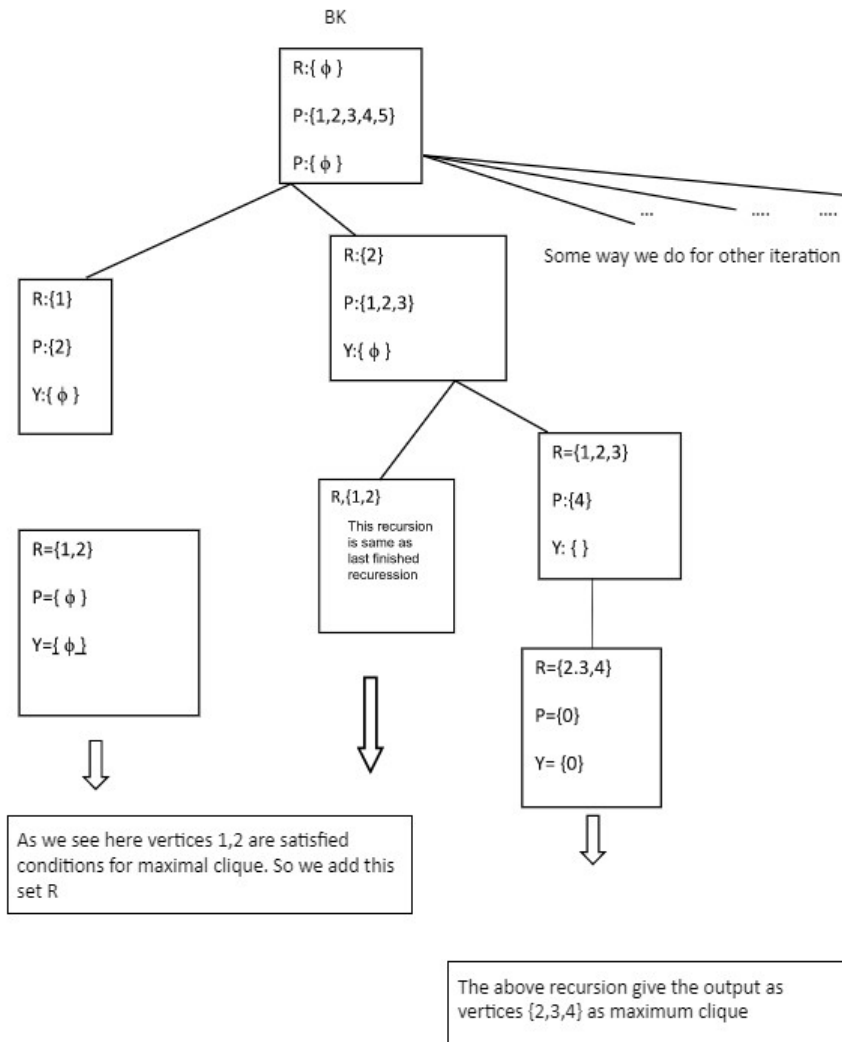


Figure 12 Work Flow of B-K Algorithm

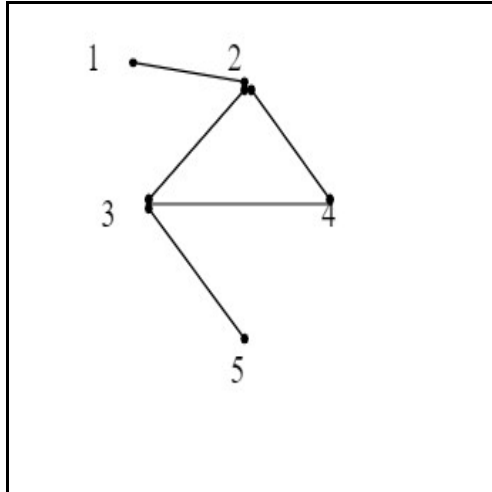


Figure 13 The final result produced by B-K algorithm

CHAPTER-5:- Experiment

This chapter presents our experiments for entity extraction on “ADR”. The experiments have been performed on drug review corpus (CADEC) that has been annotated for ADR entities. Our proposed model is MAC model using “YELPBERT”. The results have been compared 4 state of art models : “CombB model”, Graph model, “TransE model”(using ELMO embeddings), BIOES model (using ELMO embeddings. In addition to comparison with 4 SOTA models, we also want to see that whether YelpBERT embeddings used in our “MAC model” perform better than “BERT” embeddings used in TransE model for continuous and discontinuous entity extraction for ADR.

For our Experimental result we perform an in-depth analysis of experimental results in context of : continuous versus discontinuous entities (section 5.4.2) , overlapped entities (section 5.4.3), span interval of entities (section 5.4.3) which gives detailed about the performance analysis which says what are impact of overlapping of structure and how Mac model is more resilient to overlapping patterns and another is impact of gap and span length, here the “interval length” refers to the number of words among “discontinuous segments”. In addition, span length refers to the total amount of words in a span.

In Section 5.1, we will see about the data set used in this thesis, We are using “CADEC” While section 5.2 contains detailed about the implementation that how we take the corpus data set and how we preprocess the data. And how we the convert the data set according to our model required which is preprocessed data set. Then we passing this data set to BSI tagging scheme then passing the data set to the pre trained model yelpBert and finding the results on discontinuous as well continuous entities. In section 5.4, we will see the tables of results come outs after the model runs and storing results and explaining its meaning.

5.1 Datasets

The “CSIRO Adverse Drug Event Corpus” is a comprehensive annotated corpus of patient-reported “Adverse Drug Events”. This corpus is important for research on extracting information from social media, or more specifically text mining, in order to detect probable adverse medication responses from direct patient reports. [2].

We have discuss about the corpus in detailed in chapter 2 in section 2.3

In this Cadec corpus, there are 5 entities available. That is shown below in the given table.

Entity	Description	Example
Drug	This entity contains the name of a drug or medicine which are related with drugs, which include product and trade name.	Diclofenac Sodium Zipsor.
ADR	It comprises adverse drug reactions that are firmly related with a drug and are marked with an ADR label, according to the text.	Drowsiness, Diarrhea.
Disease	A description of the disease that is causing the patient to take the medicine	Arthritis, migraines.
Symptom	The reason why patient taking the drug	knee pain, lower back pain.
Finding	The symptoms or side effect which is unsure which one it belongs to, is referred to as a clinical finding.	Stroke, menopause.

Table 6 Corpus entities

5.2 Data preparation and Implementation

In this section we describe the preprocessing and preparation of our data for Mac model, followed by training and testing details. Figure 15 figure shows the flow diagram presents the preprocessing of original data set and then model implementation.

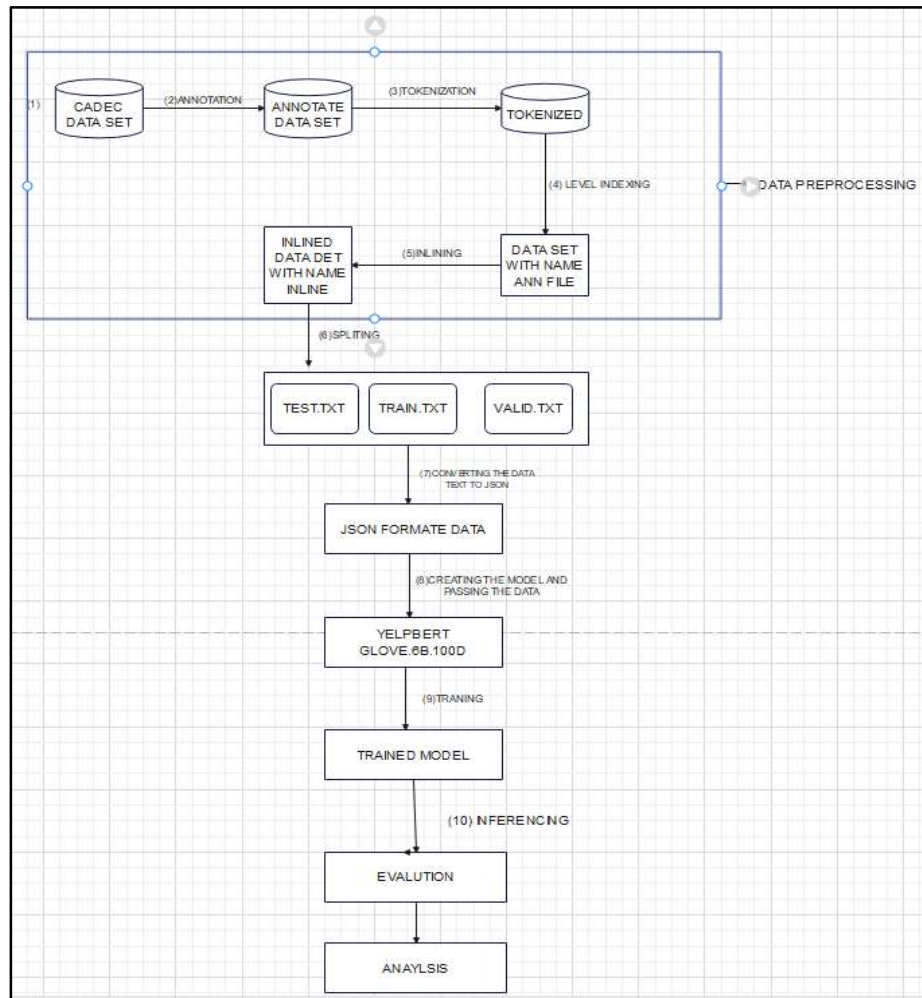


Figure 14: The flow diagram that how the original data set went under the preprocessing and then model implementation

Following is the step by step explanation of figure 15 :-

1. In the first step we annotate the data that is available in the folder original and text which is a subpart of the cadec corpus. In this step we take original and text file and it is preprocessed so that the file can be tokenized.

As for example:-

(Legs tingling in) to (tingling in legs)

(Severe joint pains) to (joint pains servers) etc.

And we store this data as a file which is ann text file.

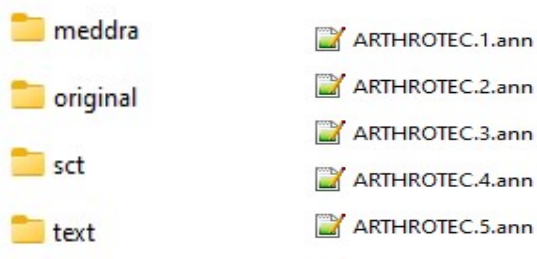


Figure 15 these are the original folders in the CADEC corpus

2. Second step is for tokenization with ann files as input. The documents are segmented into sentences and further to tokens. Total numbers of documents is 1250, 7597 sentences and 122938 tokens. Where the input is a previously generated ann file. And output is the count of the documents, sentences and tokens. The tokenized data shown in figure 17. In the figure 17, first column shows the name of the drug name (document name) whereas 2nd column shows in which line “ADR” found where 0 denotes first line. And in the 3rd column shows the entity type which is “ADR”. 4th column shows positions of “ADR” and at last column shows the name of “ADR”

ARTHROTEC.101	0	ADR 13,15	shortness of breath
ARTHROTEC.101	0	ADR 20,20	depression
ARTHROTEC.101	0	ADR 22,22	cramping
ARTHROTEC.101	0	ADR 24,25	upset stomache
ARTHROTEC.103	0	ADR 4,4	bleeding
ARTHROTEC.103	2	ADR 9,11	so much pain
ARTHROTEC.103	3	ADR 2,6	feel weak and almost fainted
ARTHROTEC.104	0	ADR 1,1	diarrhea
ARTHROTEC.104	0	ADR 4,4	constipation
ARTHROTEC.104	0	ADR 6,6	fatigue
ARTHROTEC.105	0	ADR 15,16	body swelling
ARTHROTEC.105	2	ADR 2,4	forgetfulnes and confussion
ARTHROTEC.105	4	ADR 3,4	loose stools
ARTHROTEC.105	1	ADR 0,0	Insomina
ARTHROTEC.105	3	ADR 2,6	"empty stomach" feeling
ARTHROTEC.105	0	ADR 16,16,18,18	swelling face
ARTHROTEC.105	0	ADR 16,16,20,20	swelling wrists
ARTHROTEC.105	0	ADR 16,16,22,22	swelling abdomen
ARTHROTEC.105	0	ADR 16,16,24,24	swelling thighs
ARTHROTEC.107	0	ADR 3,4	rectal bleed
ARTHROTEC.107	1	ADR 2,3	extremely sick

Figure 16 The tokenized data of original files data

3. After preprocessing in step 1 and 2 ,then we will go for the next third step where we convert annotations from character level offsets to token level id for this we gave the input is ann file which we created early. Where we find a token whose original end offset is 172 by adjusting its offset. In figure 18 is the output of this step. In this we just giving the new indexes to the “ADR” entites.

ARTHROTEC.1	ADR	9,19	bit drowsy
ARTHROTEC.1	ADR	29,50	little blurred vision
ARTHROTEC.1	ADR	62,78	gastric problems
ARTHROTEC.1	ADR	437,453	feel a bit weird
ARTHROTEC.10	ADR	0,12	Hunger pangs
ARTHROTEC.100	ADR	48,64	vaginal bleeding
ARTHROTEC.100	ADR	93,105	stomach pain
ARTHROTEC.100	ADR	107,119	canker sores
ARTHROTEC.100	ADR	133,141	headache
ARTHROTEC.101	ADR	58,77	shortness of breath
ARTHROTEC.101	ADR	90,100	depression
ARTHROTEC.101	ADR	102,110	cramping
ARTHROTEC.101	ADR	112,126	upset stomache

Figure 17 The ann file formed by adding all individuals file in original file

- At 4th step we convert the ann into inline which stores document number with medicine name, ADR position and reviews of patient. Inline file description show in figure 19.

```
Document: ARTHROTEC.1
I feel a bit drowsy & have a little blurred vision , so far no gastric problems
3,4 ADR|8,10 ADR|15,16 ADR

Document: ARTHROTEC.1
I ' ve been on Arthrotec 50 for over 10 years on and off , only taking it when

Document: ARTHROTEC.1
Due to my arthritis getting progressively worse , to the point where I am in te

Document: ARTHROTEC.1
every day for the next month to see how I get on , here goes .

Document: ARTHROTEC.1
So far its been very good , pains almost gone , but I feel a bit weird , didn '
13,16 ADR

Document: ARTHROTEC.10
Hunger pangs .
0,1 ADR
```

Figure 18 The inline file is formed by merging the original folder files and text folder files

- At final we got annotated and the span of ADR we sent that inline file for the split the data set into train, dev. And a test which is in text formatted.



```
dev.txt
test.txt
train.txt
```

Figure 19 the splitting of the data set inline which was preprocessed by CADEC data set

- As our model required a json format of data which is easy to handle so we convert this text file format to json format and we pass this json format data(train, t test, valid) to the model where we are using the pre-trained model BERT/YELPBERT . The json file format is showed in figure 21.

Figure 21 shows the json file having dictionary text which contains the review , word list contains the token,word2char conations span of words and “entity type” contains details about the ADR type , text, and char span.

```

{"text": "OK for 4 years , then muscle cramps , spasms , weakness in legs and right hand progressed to atrophy .",
 "word_list": ["OK", "for", "4", "years", ",", "then", "muscle", "cramps", ",", "spasms", ",", "weakness", "in", "legs", "and", "right", "hand",
 "word2char_span": [[0, 2], [3, 6], [7, 8], [9, 14], [15, 16], [17, 21], [22, 28], [29, 35], [36, 37], [38, 44], [45, 46], [47, 55], [56, 58],
-[64, 67], [68, 73], [74, 78], [79, 89], [90, 92], [93, 100], [101, 102]],
 "entity_list": [{"text": "muscle cramps", "type": "ADR", "char_span": [22, 35]}, {"text": "spasms", "type": "ADR", "char_span": [38, 44]}, {"t
"type": "ADR", "char_span": [47, 63]}, {"text": "atrophy", "type": "ADR",
-char_span": [93, 100]}]}

```

Figure 20 Inside dictionary form of json file

7. After completing the data formatting we need some more data preprocessing is required like adding test id ,train id, validation, char span, word span, entities list and assigning “ADR” values to text(reviewed). In doing so we also pass the data set with pre-trained model for the preprocessing which assign “ADR” tag type of text with the language English this will produce output data like train_data.json, test_data.json, valid_data.json ,dictionary_data.jon and statics.json where statistic.json will provide information of the count of the word number, character number, maximum sub words sequence length, entity type number whereas we have the total number of test data is 1160 train data is 5340 and valid data is 1097.

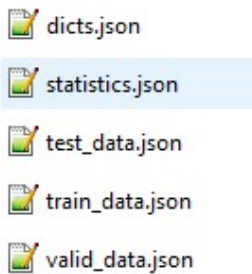


Figure 21 These files created when we perform some embeddings and preprocessed

```

{"id": "train_0", "text": "Ate a decent meal and took one pill at 4 : 30 pm , the 75 size , on 9 / 10 .",
"entity_list": [],
"features": {"word_list": ["Ate", "a", "decent", "meal", "and", "took", "one", "pill", "at", "4", ":", "30", "pm", ",", "the", "75", "s",
"word2char_span": [[0, 3], [4, 5], [6, 12], [13, 17], [18, 21], [22, 26], [27, 30], [31, 35], [36, 38], [39, 40], [41, 42], [43, 45], [
[58, 62], [63, 64], [65, 67], [68, 69], [70, 71], [72, 74], [75, 76]],
"subword_list": ["ate", "a", "decent", "meal", "and", "took", "one", "pill", "at", "4", ":", "30", "pm", ",", "the", "75", "size", ",",
"subword2char_span": [[0, 3], [4, 5], [6, 12], [13, 17], [18, 21], [22, 26], [27, 30], [31, 35], [36, 38], [39, 40], [41, 42], [43, 45]
[55, 57], [58, 62], [63, 64], [65, 67], [68, 69], [70, 71], [72, 74], [75, 76]],
"subword2word_id": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22],
"word2subword_span": [[0, 1], [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9, 10], [10, 11], [11, 12], [12, 13], [1
[18, 19], [19, 20], [20, 21], [21, 22], [22, 23]]}}

```

Figure 22 the last preprocess json file ready to inside the model

8. At this step we are going to train the model where we pass the pre-trained model, here the pre-trained model name is “BERT/YelpBert” and passing the pre-trained embedding ELMO/glove.6b.100D.txt.

9. Training of model has been done in a 7 stage according to figure 15 in the first time we load the data and the in the second stage we split the data after splitting the data we provide indexing to the data then after Tag the data .After we go for step 5 where we initiated the model at stage 6 we put the data into the data loaders and in last we set optimizer and trainer.

5.2.1 Parameter Tuning

The network parameters are optimized via way of means of Adam which has learning rate has value is 1e-5. We fixed the batch size for learning is 12. We tuned and set all hyper parameters on the dev sets. We perform our trials on NVIDIA Quadro P4000 GPU for at most 30 epochs and select the model with the best performance on the dev set to output results on the test sets. Select the version with the great overall performance at the dev set to output outcomes at the take a look at set. Select the version with the great overall enactment at the dev set to output outcomes at the take a look at set. We record the take a look at rating of the run with the median dev rating amongst five randomly initialized runs.

Once we trained the model we will go for the Evaluation of the model. This will generate data which train and test. At last we go for the Analysis part, where we discuss our results and graph of the models.

5.3 Evaluation Parameters

Following are the evaluating parameters for analyzing our result “confusion matrix”, “recall”, “precision”, and f-measure.

“Confusion Matrix” is a special table that allows the evaluation of an algorithm's performance. It contains the data about actual and predicted classes done by a classification organization.

Model\Actual	Correct	Not Correct
Selected	TP	FP
Not selected	FN	TN

Confusion Matrix

Table 7 Confusion Matrix

TP (True Positive) → correctly selected by model

FN (False negative) → not selected by model but actually correct

TN (True Negative) → not selected by model and not correct

FP (False Positive) → wrongly selected by model

Precision: Precision evaluation metric is used to evaluate the performance for classification, information retrieval, etc tasks. Precision can be well-defined as the numeral of related documents retrieved out of a total number of documents retrieved. Precision processes the accuracy of the result obtained.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall measures the coverage of the result obtained.

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

In many cases both recall and precision may not be high simultaneously in other words in most of the cases there has to be a compromise between recall and precision. Therefore, recall and precision provides different views of evaluation. In order to judge the quality of result based on both “precision” and recall, F-measure can be used. “F-measure” is the harmonic mean of recall and precision.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.4. Experimental Results

Table 8 present the statistical of traning testing and validation data set. In table 8, “S represent the number of sentences M represent the number of sentences, and D discontinuous mentions of sentences. P denotes the percentage of discontinuous mentions in total mentions”. As in first column contains data for training which contains 5,340 sentences, 4,430 mentions entities and 491 discontinuous entries and last row 11.1 is the total percentage of discontinuous entities in all train data set.

CADEC			
	train	test	dev
S	5,340	1,160	1,097
M	4,430	990	898
D	491	94	94
P	11.1	9.5	10.5

Table 8 “Statistics of datasets. S, M, and D respectively represent the number of sentences, total mentions, and discontinuous mentions. P denotes the percentage of discontinuous mentions in total mentions”.

5.4.1 Over All Results

Table 9 presents results of all the models based on “precision”, “recall” and “f measure” of all four SOTA models including Mac model and its comparisons which shows that Mac model work better than the other model (“BIOE”, “Graph”, “CombB”, “TransE”)

CADEC			
Model	Precision	Recall	F1
BIOE	68.7	66.1	67.4
Graph	72.1	48.4	58
CombB	69.8	68.7	69.2
TransE	68.9	69.0	69
Mac	70.8	74.9	74

Table 9 “Main results on three benchmark datasets. Bold marks highest number among all models”

5.4.2 Result Comparison for Continuous and Discontinuous Entities

As it has been mentioned in our work that discontinuous entity extraction is more challenging than continuous entity extraction, our aim was also to compare the performance of various models on continuous and discontinuous entities separately.

Table 10 contains the results for mentions of “discontinuous entities”. Two scores are listed in the Table, separated by a slash (“/”). The former represents the grade for sentences that have at least one “discontinuous component”. The latter score only takes references of discontinuous entities into account.

CADEC			
Model	Precision	Recall	F1
BIOE	68.3 /50.8	52.0/1.0	57.3/1.8
Graph	69.5/ 60.8	43.2/14.8	53.3/23.9
CombB	63.9/44.0	57.8/23.4	60.7/30.6
TransE	66.5/41.2	64.3/35.1	65.4/37.9
Mac	72.4/52.9	66.7/36.3	67.8/43.4

Table 10 “Results for mentions of discontinuous entities. Two scores are listed in the Table, separated by a slash (“/”). The former represents the grade for sentences that have at least one discontinuous component. The latter score only takes references of discontinuous entities into account”

Based on the Experiments, the results can be summarized as follows:

- (1) This method, Mac, performs significantly better than all other approaches and reach “SOTA” F1 score in all 3 datasets.

(2) “BERT based Trans model” achieves worse results than “ELMo-based” model purported equivalent on the original paper.[2]

(3) Via SOTA method TransE, Mac, gets major improvements In F1, an average of 2.6% scores across all three datasets. Moreover, The Wilcoxon test is important because there is ($p < 0:05$) difference between our model and TransE. We think it's because of TransE. As it introduces, it is a multi-stage method by its nature. Few dependent actions, therefore the problem of exposure bias. For our Mac method, gracefully parses discontinuous NER divides the job into two self-determining subtasks and learns them together with a common model that achieves consistency education and conclusion.

(4) Can be CombB seen as approximately our pipeline form technique reaffirms performance gaps efficiency of our single-stage learning outline. As in Table 8, only about 10% are mentioned as “discontinuous” in all three distant datasets a smaller amount than perpetual existence speaks of. To evaluate efficiency of our proposed model on recognition “Discontinuous” words after muis and Lu (2016)[16], we report the results in sentences containing at least one interrupted word. We also report estimate results only when intermittent is considered to be mentioned. Points in these two settings are disconnected by a slash in Table 10. Comparing Tables 9 and 10, we can see that: “BIOE model” outperforms Graphics as soon as testing on the full dataset, but much worse speaks inconsistently. Steadily, our model beat the base models once again in terms of F1 Point. Although more or less models perform better than "Mac" in terms of precision or recall, they severely degrade one another's score, which results in a lower F1 score than "Mac".

5.4.3 IMPACT OF OVERLYING STRUCTURE

As we discuss in the introduction and as well in methodology, that overlapping is a very collective problem in the “discontinuous entity” mentions which we overcome by using maximal clique techniques. So to estimate the capability of introducing the “Mac model” on mining overlying structure, as information is provided in paper Dai et al [2]. We are dividing the test file into sets of 4 categories.

- 1.) ‘No overlapping’
- 2.) ‘Left overlapping’
- 3.) ‘Right overlapping’

4.) ‘Multiple overlapping’

	Texts	Mentions
No Overlapping	Back pain and produce sputum	Back pain and produce sputum
Left Overlapping	Pain in head and spinal	[pain] in head [pain] in spinal
Right Overlapping	Lower back and shoulder pain	Lower back[pain] Shoulder [pain]
Multiple Overlapping	Cough with yellow or bloody sputum	[cough]with yellow [sputum] [cough]with bloody [sputum]

Table 11 “Shows left overlapping ,right overlapping , no overlapping and Multiple Overlapping”

We provide an example in Table 11 where it shows what is “no overlapping “,”left overlapping” , “right overlap” and “multiple overlap”. Now we have come to Section where we see the performance of Mac and TransE on all the overlying patterns. TransE sometimes scores zero on some patterns. It is possible that it is the consequence of inadequate training due to the fact that some overlying designs have comparatively fewer samples in the training sets which is shown in table 12, where the sequence operation configuration of the transition based model is rather greedy for data.

Patterns	CADEC		
	Train	Dev	Test
No Overlapping	57	9	16

Left Overlapping	270	54	41
Right Overlapping	113	16	23
Multiple Overlapping	51	15	14

Table 12 Results of overlapping In CADEC Corpus

By constraint it is found that Mac model is more resistant to overlapping patterns and we believe that this is due to one of dual design choice :-

- 1.) The “grid tagging” system has solid in properly recognizing overlying segments and combining them into a “segment graph”.
- 2.) On the created graph the “maximal clique” approach can successfully improve all of the contestant overlying entity mentions.

Figure 24 is comparison outcomes of the two model “Mac” model and “TransE” model where it is clear from bar graph Mac Model work efficiently then “TransE” model. We can see when there is no over lapping “TransE” model work good but when the overlapping comes in entities Mac model perform better then the “TransE” model.

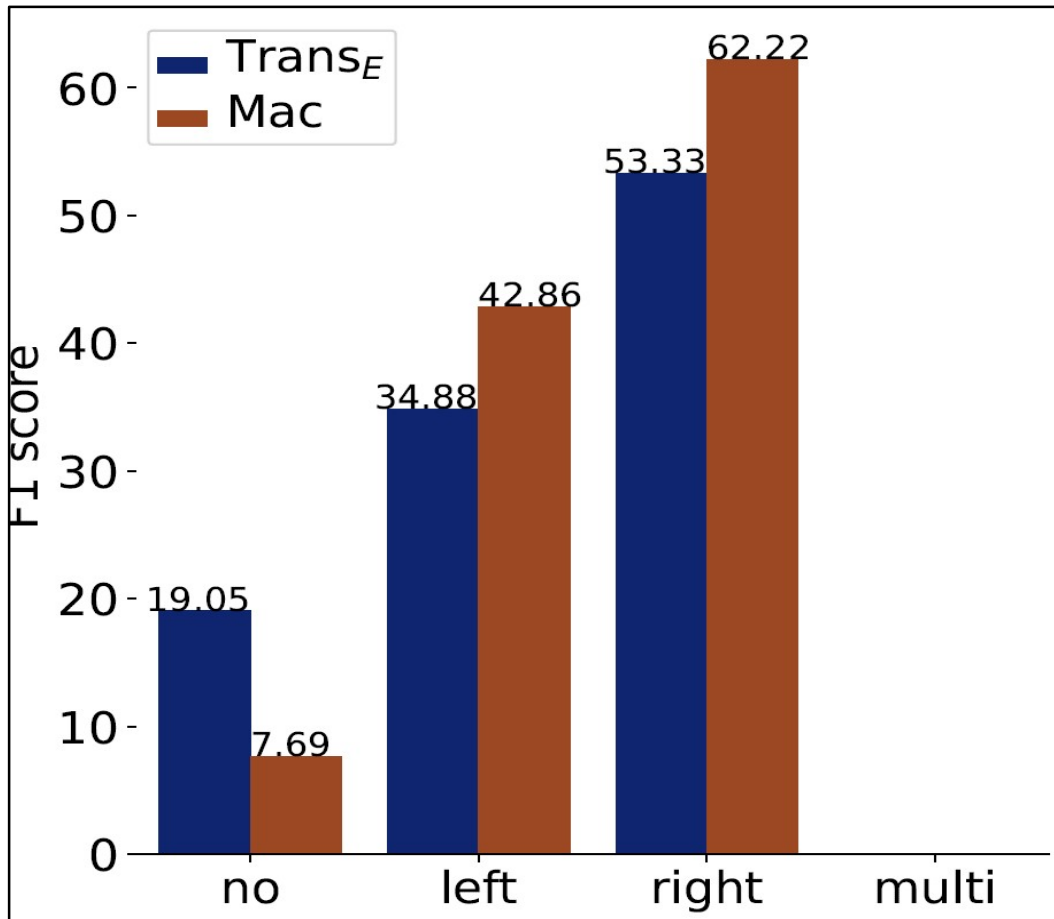


Figure 23 “The comparison between TransE and Mac model Performance on different overlapping patterns”

5.4.4 IMPACT OF INTERVAL AND SPAN LENGTH

The overall length of a “discontinuous” comment is often longer than the length of a continuous one because of the intervals that exist between the segments. When each component is considered, the overall span becomes even more extensive. In other words, the multiple words that make up a discontinuous mention may be spatially isolated from one another, which makes the work of NER for discontinuous mentions more difficult than NER for conventional mentions. We analyze the results of test sets conducted proceeding a variety of intervals and span lengths so that we may have a better understanding of how robust Mac is in a variety of environments. The length of words that exist between two consecutively discontinuous segments is referred to as the interval length. The length of words that make up the whole span is referred to as the span's length. Here is an example, for the entity to mention “Extreme shoulder pain” in the patient review “Extreme neck, shoulder

and lower back pain” the “interval length = 5” and the “span length = 8”. Such phenomena demand models with the capacity of getting the semantic dependence between the distance segments. In order to make the analysis process more straightforward, we have presented the distribution of all dataset’s interval and span lengths in Tables 13 and 14, respectively. In addition, Figure 25 and 26 presents the F1 scores obtained by TransE and Mac for a variety of interval and span lengths. As can be seen, Mac performs better than TransE in the majority of settings. In spite of the fact that Mac has been shown to be inferior in some situations, the number of examples in which this has occurred is insufficient to refute Mac's preeminence. For instance, on CADEC, TransE performs better than Mac when the span length is 8, despite the fact that there are only 10 samples included in the test set. We find out something interesting: When interval length = ‘1’ and span length = ‘3’, both Mac and TransE don't do well, even though the training examples are enough (see length is equal to 1 in Table 13 and length is equal to 3 in Table 14). These could be effect from two things:

(1) Even though there are enough training examples, their features and contexts are unlike from those in the test set.

(2) “Discontinuous mentions” with an interval length of 1 are harder than the others because there is only one word between each segment. This makes these discontinuous mentions look a lot like continuous mentions, which makes the model think they are continuous mentions. We'll deal with this problem when we get to it.

Length	CADEC		
	train	dev	test
=1	36	8	8
=2	217	42	54
=3	56	14	12
=4	68	14	8
=5	36	4	4
=6	30	3	3

>=7	48	9	5
-----	----	---	---

Table 13 Statistics of interval length

Length	CADEC		
	train	dev	test
=3	10	3	4
=4	95	23	24
=5	67	13	15
=6	91	13	16
=7	57	15	9
=8	53	9	10
>=9	118	18	19

Table 14 Statistics of span length

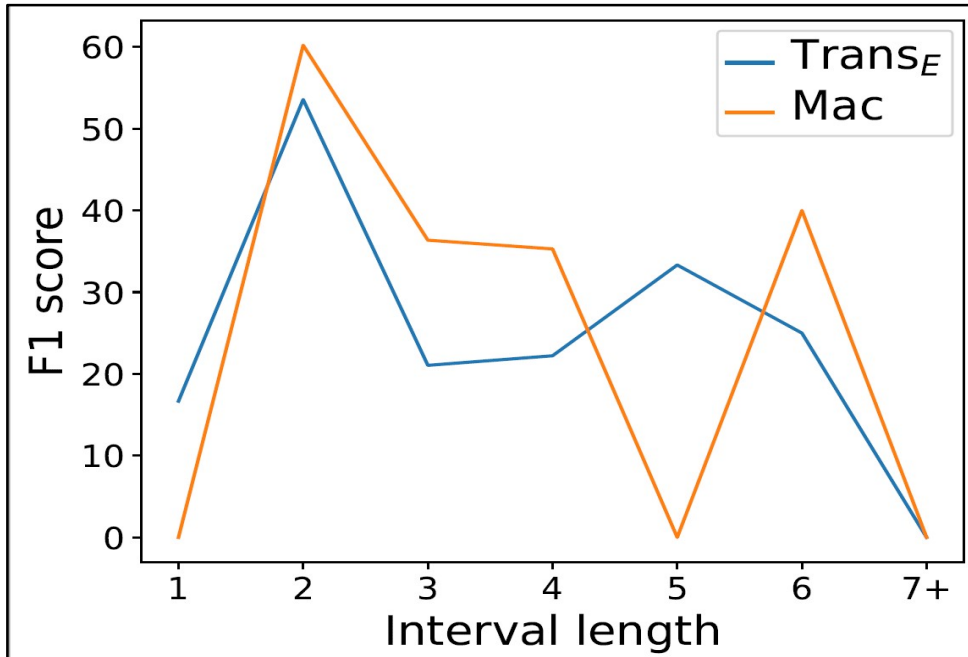


Figure 24 “The comparison between TransE and Mac model Performance on different interval length patterns”

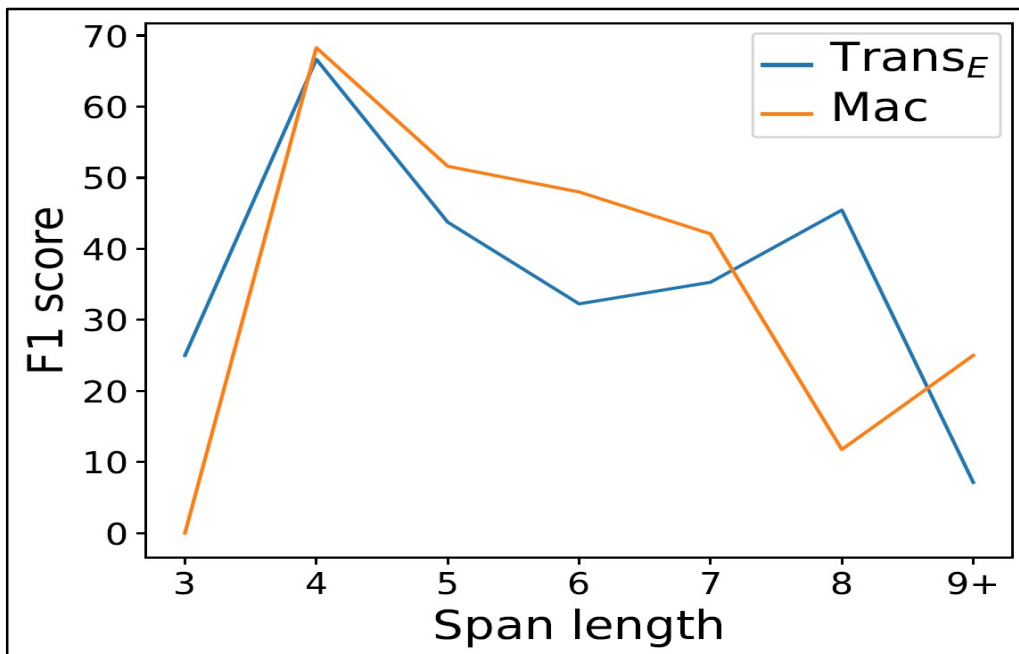


Figure 25 comparison between TransE and Mac model Performance on different span length patterns

Chapter 6: Conclusion and Future Work

We introduce a simple effective “transition based model Mac”, which can identify discontinuous entities. We first create a segment graph for every sentence in which every node denotes a segment (A Continuous entity on its very own sort of discontinuous entity) and link the edge between the two notes that belong to the same entity. We construct “Discontinuous NER” as the job for identifying maximal cliques in a segment graph. In this “Mac Model”, It divides the construction of the segments graph into two distinct self-determining “2-D grid tagging problems”, and solve them parallel in one step. Solving the exposure bias issues in the earlier works in a subtle manner. We did experiments on a benchmark data set named CADEC (CSIRO Adverse Drug Event Corpus). Mac models perform good as compared to the other four model which is “TransE model”, ”Graph model”, ”CombB model”, “BOIE model” which are related to identifying the discontinuous entities like SOTA model. Mac model is efficient to identifying the overlapping and discontinuous entities with distributing the accuracy of continuous entries.

Further study indicates the capabilities of our approach in distinguishing discontinuous. And also indicates the capabilities of our approach on overlapping entity references by using maximal cliques’ techniques. In the upcoming, we would want to investigate alike constructions in generally other info mining tasks, mainly such as event extraction and nested NER in a subtle approach.

REFERENCES

- [1] “Cadec: A corpus of adverse drug event annotations | Elsevier Enhanced Reader.” <https://reader.elsevier.com/reader/sd/pii/S1532046415000532?token=979101B8FA4F4EC7BC34513EFD1C0FCA928DB584A7925311E36787CF118C7A2EB1F396D81BC5EFE11940842FB631B742&originRegion=eu-west-1&originCreation=20220718101519> (accessed Jul. 18, 2022).
- [2] X. Dai, S. Karimi, B. Hachey, and C. Paris, “An Effective Transition-based Model for Discontinuous NER,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Jul. 2020, pp. 5860–5870. doi: 10.18653/v1/2020.acl-main.520.
- [3] A. O. Muis and W. Lu, “Learning to Recognize Discontiguous Entities,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, Nov. 2016, pp. 75–84. doi: 10.18653/v1/D16-1008.
- [4] W. Zhang, Y. Feng, F. Meng, D. You, and Q. Liu, “Bridging the Gap between Training and Inference for Neural Machine Translation,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul. 2019, pp. 4334–4343. doi: 10.18653/v1/P19-1426.
- [5] Y. Wang, B. Yu, H. Zhu, T. Liu, N. Yu, and L. Sun, “Discontinuous Named Entity Recognition as Maximal Clique Discovery,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online, Aug. 2021, pp. 764–774. doi: 10.18653/v1/2021.acl-long.63.
- [6] B. Tang, Y. Wu, M. Jiang, J. C. Denny, and H. Xu, “Recognizing and Encoding Disorder Concepts in Clinical Text using Machine Learning and Vector Space Model*.”
- [7] A. Stubbs and Ö. Uzuner, “Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus,” *J. Biomed. Inform.*, vol. 58, pp. S20–S29, Dec. 2015, doi: 10.1016/j.jbi.2015.07.020.

- [8] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, “2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text,” *J. Am. Med. Inform. Assoc.*, vol. 18, no. 5, pp. 552–556, Sep. 2011, doi: 10.1136/amiajnl-2011-000203.
- [9] L. Chen *et al.*, “Extracting medications and associated adverse drug events using a natural language processing system combining knowledge base and deep learning,” *J. Am. Med. Inform. Assoc. JAMIA*, vol. 27, Oct. 2019, doi: 10.1093/jamia/ocz141.
- [10] F. Christopoulou, T. T. Tran, S. K. Sahu, M. Miwa, and S. Ananiadou, “Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods,” *J. Am. Med. Inform. Assoc.*, vol. 27, no. 1, pp. 39–46, Jan. 2020, doi: 10.1093/jamia/ocz101.
- [11] *Artificial Intelligence in Medicine*. Accessed: Jul. 18, 2022. [Online]. Available: <https://link.springer.com/book/10.1007/978-3-319-59758-4>
- [12] Y. Wang *et al.*, “Clinical information extraction applications: A literature review,” *J. Biomed. Inform.*, vol. 77, pp. 34–49, Jan. 2018, doi: 10.1016/j.jbi.2017.11.011.
- [13] L. Chiticariu, Y. Li, and F. R. Reiss, “Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, Oct. 2013, pp. 827–832. Accessed: Jun. 25, 2022. [Online]. Available: <https://aclanthology.org/D13-1079>
- [14] S. Wu *et al.*, “Deep learning in clinical natural language processing: a methodical review,” *J. Am. Med. Inform. Assoc.*, vol. 27, no. 3, pp. 457–470, Mar. 2020, doi: 10.1093/jamia/ocz200.
- [15] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, *Linguisticae Investigationes* 30 (2007) 3–26, <https://doi.org/10.1075/li.30.1.03nad>.
- [16] Y. Zhang, J. Xu, C. Hui, J. Wang, Y. Wu, M. Prakasam, X. Hua, Chemical named entity recognition in patents by domain knowledge and unsupervised feature learning, *Database J. Biol. Databases Curat.* 2016 (2016), <https://doi.org/10.1093/database/baw049>

- [17] R.T. McDonald, F. Pereira, Identifying gene and protein mentions in text using conditional random fields, *BMC Bioinf.* 6 (2005) 1–7, <https://doi.org/10.1186/1471-2105-6-S1-S6>.
- [18] S. Saha, A. Ekbal, U.K. Sikdar, *Named Entity Recognition and Classification in Biomedical Text Using Classifier Ensemble*, Inderscience Publishers, 2015.
- [19] Q. Wei, T. Chen, R. Xu, Y. He, L. Gui, Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks, *Database* 2016 (2016), <https://doi.org/10.1093/database/baw140>
- [20] D. Zeng, C. Sun, L. Lin, B. Liu, LSTM-CRF for drug-named entity recognition, *Entropy* 19 (2017), <https://doi.org/10.3390/e19060283>.
- [21] L. Derczynski, I. Augenstein, K. Bontcheva, USFD: Twitter NER with Drift Compensation and Linked Data, *arXiv: Computation and Language*, 2015, 48–53. <http://doi.org/10.18653/v1/W15-4306>.
- [22] H. He, X. Sun, F-score driven max margin neural network for named entity recognition in Chinese social media, In: *Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2, 2017, pp. 713–718. <http://doi.org/10.18653/v1/e17-2113>.
- [23] Aramaki, E., Miura, Y., Tonoike, M., Ohkuma, T., Masuichi, H., Waki, K., and Ohe, K. (2010). Extraction of adverse drug effects from clinical records. In *MedInfo*, pages 739743.
- [24] A. Cami, A. Arnold, S. Manzi, and B. Reis, “Predicting Adverse Drug Events Using Pharmacological Network Models,” *Sci. Transl. Med.*, vol. 3, no. 114, Dec. 2011, doi: 10.1126/scitranslmed.3002774.
- [25] D.-S. Cao et al., “Integrating Multiple Evidence Sources to Predict Adverse Drug Reactions Based on a Systems Pharmacology Model,” *CPT Pharmacomet. Syst. Pharmacol.*, vol. 4, no. 9, pp. 498–506, 2015, doi: 10.1002/psp4.12002.
- [26] S. Dey, H. Luo, A. Fokoue, J. Hu, and P. Zhang, “Predicting adverse drug reactions through interpretable deep learning framework,” *BMC Bioinformatics*, vol. 19, no. S21, p. 476, Dec. 2018, doi: 10.1186/s12859-018-2544-0.

- [27] J. Hakenberg et al., “A SNPshot of PubMed to associate genetic variants with drugs, diseases, and adverse reactions,” *J. Biomed. Inform.*, vol. 45, no. 5, pp. 842–850, Oct. 2012, doi: 10.1016/j.jbi.2012.04.006.
- [28] E. Pauwels, V. Stoven, and Y. Yamanishi, “Predicting drug side-effect profiles: a chemical fragment-based approach,” *BMC Bioinformatics*, vol. 12, no. 1, p. 169, Dec. 2011, doi: 10.1186/1471-2105-12-169.
- [29] S. Lee, K. H. Lee, M. Song, and D. Lee, “Building the process-drug–side effect network to discover the relationship between biological Processes and side effects,” *BMC Bioinformatics*, vol. 12, no. S2, p. S2, Dec. 2011, doi: 10.1186/1471-2105-12-S2-S2.
- [30] I. Wallach, N. Jaitly, and R. Lilien, “A Structure-Based Approach for Mapping Adverse Drug Reactions to the Perturbation of Underlying Biological Pathways,” *PLoS ONE*, vol. 5, no. 8, p. e12063, Aug. 2010, doi: 10.1371/journal.pone.0012063.
- [31] S. Dasgupta, A. Jayagopal, A. L. Jun Hong, R. Mariappan, and V. Rajan, “Adverse Drug Event Prediction Using Noisy Literature-Derived Knowledge Graphs: Algorithm Development and Validation,” *JMIR Med. Inform.*, vol. 9, no. 10, p. e32730, Oct. 2021, doi: 10.2196/32730.
- [32] D. M. Bean et al., “Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records,” *Sci. Rep.*, vol. 7, no. 1, p. 16416, Dec. 2017, doi: 10.1038/s41598-017-16674-x.
- [33] P. Ernst, A. Siu, and G. Weikum, “KnowLife: a versatile approach for constructing a large knowledge graph for biomedical sciences,” *BMC Bioinformatics*, vol. 16, no. 1, p. 157, Dec. 2015, doi: 10.1186/s12859-015-0549-5.
- [34] [Welcome to MedDRA | MedDRA](#)