Adverse Drug Reaction (ADRs) Extraction Using Transfer of Learning

Submitted by
Sajid Hussain

Supervisor: Dr. Hammad Afzal

A thesis submitted to the faculty of Computer Software engineering, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Computer Science (Software) Engineering

2020

Abstract

Adverse Drug Reactions (ADRs) are significantly harmful for health. Existing studies utilize traditional and deep learning techniques to detect ADRs from the given text. Bidirectional Encoder Representations from Transformers (BERT) overcame the predominant neural networks bringing remarkable performance gains. However, training BERT is computationally expensive which limits determining the most important hyper parameters for the downstream task. Furthermore, developing an end-to-end ADR extraction system comprising two downstream tasks i.e. text classification for filtering text containing ADRs and extracting ADR mentions from the classified text is also challenging. In this work, we present an end-to-end system for modelling ADR detection from the given text by ne-tuning BERT with a highly modular Framework for Adapting Representation Models (FARM). FARM provides support for multi-task learning by combining multiple prediction heads which makes training of the end-to-end systems easier and computationally faster. In the proposed model, one prediction head is used for text classification and another is used for ADR sequence labelling. The model is fine-tuned on the data collected from Twitter and PubMed abstracts. The proposed model is compared with the state-of-the-art techniques and it is shown that it yields better results for the given task.

Keywords: Multitask learning, Fine-tuning, BERT, FARM, ADR

Dedication

The thesis is devoted to

MY BELOVED FATHER

You will always be remembered

Acknowledgments

Starting with the greatest name of Allah, most gracious and most merciful. His blessings are unbounded, his benevolence is everlasting. My all the prayers are to Allah who gave me patience and strength to overcome all my problems. Special gratitude to my supervisor, Dr. Hammad Afzal who gave me this opportunity to work on this project and for being patient enough. I will also like to pay regards my co-supervisors Dr. Naima Iltaf and Asst. Prof Bilal Rauf for guiding me throughout this project. I am so grateful to my family and friends for being there for me.

Table of Contents

Abstract I

Dedication II

Acknowledgments III

Table of Contents IV

List of Equations V

List of Figures VI

List of Tables VII

Chapter 1 1

Introduction 1

Chapter 2 5

Literature Review 5

Chapter 3 9

Proposed Methodology: FARM-BERT 9

3.1 BERT 10

3.2 Input Representation 10

3.3 Pretrained BERT 11

3.4 Fine Tuning BERT with FARM for ADR Detection 12

3.5 ADR Prediction 13

3.6 Optimization 13

Chapter 4 14

Experiments and Results 14

4.1 Datasets 14

4.2 Evaluation Metrics 16

4.3 Proposed Model Configuration 16

4.4 Comparison with Baseline Models 17

4.5 Comparison of computational performance of FARM-BERT with BERT 21

4.6 Comparison with State-of-the-Art Works 24

Chapter 5 29

Conclusion 29

Chapter 6 30

References 30

List of Equations

Equation-1 11*Equation-2* 13*Equation-3* 13*Equation-4* 16*Equation-5* 16*Equation-6* 16

List of Figures

Figure 1 8

Figure 2 20

Figure 3 21

Figure 4 23

Figure 5 25

List of Tables

Table 1 *17*

Table 2 *20*

Table 3 *21*

Table 4 *22*

Table 5 *24*

Chapter 1
Introduction

Adverse Drug Reactions (ADRs) have harmful effects on health. ADR ac-cording to the definition of the World Health Organization (WHO) is a response to noxious medication which occurs as a result of normal doses used in man for diagnosing or curing a disease [1]. ADRs greatly affect quality of life and in worse cases can be a cause of death. A study showed that 3.5% of the patients were hospitalized because of ADRs [2]. It has been estimated ADRs were responsible for approximately 197,000 deaths annually in Europe [3]. The safety of a drug is monitored by the Food and Drug Administration (FDA) after its release. These surveillance activities, however, are largely reliant on a passive spontaneous reporting database known as Adverse Event Reporting System (AERS) [4]. Delayed and underreported events can make these systems inefficient.

To address the limitations of passive surveillance, active pharmacovigilance techniques used for labeling ADRs analyze frequently updated sources of data. Data from social media particularly twitter because of its public nature and vast reach can be used as a source of carrying out post-market drug surveillance. Studies have observed significant correlations between ADRs reported in AERS and those mentioned in Twitter [5]. Several studies have been conducted on Twitter data [6, 7]

however, limitations arise due to informal language of social media. As compared to Twitter, very formal description is found in biomedical text. Hence, some studies use biomedical text collected from PubMed abstracts for ADR extractions [8, 9], while some utilize data from both social media and biomedical text [10, 11]. In this work, we also use datasets from both the sources i.e. Twitter and PubMed.

ADR extraction has been performed using conventional machine learning models such as Support Vector Machine (SVM) [12], Random Forest (RF) [13], and Conditional Random Field (CRF) [14]. These models depend upon manual feature engineering. Most common features utilized by these models include n-grams, negated contexts, and semantic types from Unified Medical Language Sys- tem (UMLS), Part of Speech (POS) tags, drug names, and lexicon based features, and word embeddings [15]. Numerous studies utilize deep learning techniques such as Bidirectional Long-Short Term Memory (BLSTM) [11], Convolutional Neural Network (CNN) [16], and attention based deep neural networks [10]. Most recent studies have employed Bidirectional Encoder Representations from Transformers (BERT) and its different variants which significantly improved the performance of ADR detection [7, 17]. However, training these models is computationally expensive which limits the tuning of hyper parameters. Hence, determining the most contributing hyper parameters becomes challenging. Furthermore, ADR extraction from social media data firstly requires text classification to remove noise and filter text with ADR mentions. Text classification is

then followed by the task of ADR sequence labelling. Hence, a framework with the support of multi-task learning is needed for end-to-end modelling of the problem.

In this work, we use BERT fine-tuned via a novel framework FARM to detect ADRs on Twitter and PubMed datasets. FARM has a modular design for language models and prediction heads which makes transfer learning simpler. FARM is an adaptive model that provides support for combining multiple prediction heads on top of the language model. We present an end-to-end solution for ADR extraction by using two prediction heads with BERT; one for classifying text with ADR mentions and the other for labelling ADR sequences in the classified text. Moreover, FARM supports parallelized processing which makes learning computationally faster. The hyper parameters used in the standard BERT model are modified with FARM-BERT such that they best t the learning task. In short, primary contributions of this work are listed below:

- A novel end-to-end model FARM-BERT based on highly modular design is proposed to detect ADRs
- FARM-BERT is fine-tuned with different set of hyperparameters as compared to the standard BERT

Comparison of results shows that BERT ne-tuned using FARM outperforms state-of-the-art techniques used for extracting ADRs.

The rest of the paper is structured as follows: Section 2 presents the literature review, Section 3 proposes a framework for end-to-end detection of ADRs, Section 4 discusses experiments and results while Section 5 draws the conclusion.

Chapter 2
Literature Review

There has been a considerable amount of work for detecting ADRs from biomedical text automatically using machine learning approaches. Earlier works utilize traditional machine learning approaches with manual feature engineering. Liu et al. [12] passed a bag of words, bigrams and Part of Speech (POS) tags as features to SVM where the bag of words produced the best results. Bag of words approach is based on occurrences of words in a corpus. It ignores the semantics and syntactic of the text. Hence, this approach is not a reliable approach leading to false classifications. Alimova et al. [18] fed SVM and Logistic Regression (LR) with features including lexicon based features, sentiment features, semantic features, and word embeddings. Since, lexicon is based on a particular list of drugs, lexicon based features do not play a significant role in ADR identification. Sentiment and word embedding features have been found to be the most effective. Sarker et al. [19] used SVM fed with topic model features in combination with other features such as n-grams, sentiword scores, lexicon features, syn-set expansion features, UMLS semantic types etc. Bian et al. [20] also used semantic features based on UMLS in combination with other textual features. In the shared task Social Media Mining for Healthcare (SMM4H) 2017, the best performing system employed SVM fed with different domain specific, surface-form and sentiment features [21]. Aramaki et al. [22] used SVM and CRF for extracting adverse drug effects

using lexicon based features, POS tags, word chain etc. CRF has also been used in [14] which utilized contextual features, word embedding features, and dictionaries. Another approach [13] uses RF models fed with n-gram features, negation, sentiment etc. Traditional approaches rely upon manual feature engineering which needs considerable effort and time.

Recent approaches for ADR detection employ deep neural networks. CNN initialized with Pyysalo's word embeddings [23] has been used in [16] to detect ADRs. Huynh et al. [24] proposed Convolutional Recurrent Neural Network (CRNN) for ADR detection. In [11], BLSTM network was used with word embeddings as input features. In [24] a multi-task encoder-decoder framework has been proposed that provides end to end solution by modelling three ADR detection tasks i.e. classification of ADRs, ADR labeling and indication labelling.

To tackle the problem of limited labelled data for ADR, Gupta et al. [25] proposed a semi-supervised approach based on co-training which can augment the labelled data with large amounts of unlabeled data. Semi supervised model was also proposed in [26]. For the unsupervised learning stage, drug name was predicted on the basis of its context in the given tweet using BLSTM model. The BLSTM model initiated with word2vec based word embeddings was trained in a supervised learning stage to predict the sequence labels in tweets. Zhang et al. [27] presented a weakly supervised CNN-LSTM model to identify ADRs. Weakly labelled data was employed to pertain to the model. The model parameters were further fine-tuned on the labelled dataset.

Some models combine deep neural networks with traditional models such as BLSTM-CRF for sequence labelling [28]. They exploit both word embedding based features and other natural language processing features such as spelling features, n-gram features and POS features.

Another BLSTM-CRF model uses character embeddings in addition to word embeddings [29]. In [30, 31] combination of CNN, LSTM and CRF has been proposed where word embeddings are augmented using character level CNN. Neural network models, when processing long texts, suffer from the problem of vanishing gradient. The problem can be dealt with using an attention mechanism. In the attention mechanism, the decoder retrieves selective information from the most relevant parts of the source sentence instead of using all the information encoded into a fixed sized vector [32]. Ramamoorthy et al. [8] proposed self-attention based BLSTM model for facilitating intra-sequence interaction in the given text sequence. Ding et al. [10] proposed embedding level attention mechanism in Bidirectional Gated Recurrent Unit (BGRU) to allow the model to learn the most important features. The recent meeting of SMM4H held in 2019 showed further improvements in neural network techniques used for ADR detection [33]. Convolutional and recurrent neural architectures fed with word2vec or glove embeddings being the most popular architectures for tackling the task in 2018 were overtaken in 2019 by neural networks that used word embeddings pertained with BERT [34]. The approach of the winning team was based on retraining BERT on a large unlabeled tweets dataset collected from twitter using a list of drug

names [35]. In [7] domain specific preprocessing and an ensemble of different BERT implementations i.e. general BERTLARGE, domain specific BioBERT [36] and domain specific ClinicalBERT [37] have also been shown to be effective for ADR classification on social media. Li et al. [17] integrated BERT with CNN and utilized emotional information to distinguish between ADR and non-ADR tweets. Aroyehun et al. [6] used LSTM fed with a combination of three types of embeddings i.e. character embeddings, glove embeddings and BERT embeddings to detect ADR reportage in tweets.

Chapter 3

Proposed Methodology: FARM-BERT

We use BERT implemented via a novel framework FARM to detect ADRs. This section briefly discusses the architecture of BERT followed by a brief description of the pertained BERT used in our study. We then describe the fine-tuning of BERT with FARM. Figure 1 presents the overall architecture of the proposed system.

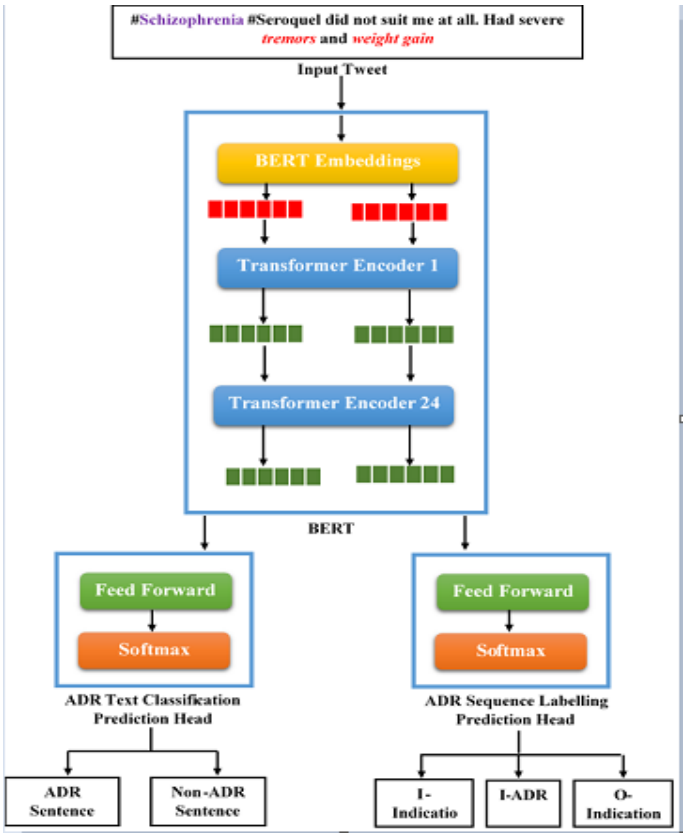


Figure 1

3.1 BERT

Training BERT involves two phases i.e. pretraining and fine-tuning. In the first phase i.e. pretraining, unlabeled data is used to train the model over different tasks. In fine-tuning, the pretrained parameters are fine-tuned on a labelled dataset to model a downstream task. The architecture of BERT is based on bidirectional transformers in multiple layers [38]. In this work, we use a BERT base which consists of 12 layers denoted as L, 768 hidden units denoted as H, and 12 self-attention heads denoted as A.

3.2 Input Representation

BERT generates contextualized embeddings. Many models have widely been used to convert words into embeddings such as word2vec, fasttext, and glove. However, these models generate embeddings of a word without considering its context. In natural language, meanings of a similar word may vary in different contexts. Context dependent representation is not captured by these models resulting in the similar vector representations of a word having different meanings in different contexts. As opposed to the previous models, BERT generates contextualized embeddings.

BERT takes as input a single sentence or a pair of sentences. BERT uses WordPiece model to tokenize the input sequence. Special tokens are added by the tokenizer at

the beginning and end of the input sequence. The first token that marks the beginning of every input sequence is [CLS]. Two sentences in the input sequence are divided by a special token [SEP]. Besides tokenizing the input sentences into words, individual words, if not found in the vocabulary, are also tokenized into subwords and characters. In this way, BERT generates embeddings for out of vocabulary words by generating embeddings of their constituent subwords and characters found in the vocabulary. In addition to producing the token embeddings, BERT generates sentence embeddings by adding embedding to each token in the tokenized text indicating whether the token belongs to the first or the second sentence. It further generates position embeddings indicating the position of a token in the input sequence. Finally, the input representation for a given token can be represented by concatenating its corresponding token embeddings, sentence embeddings and position embeddings. Let t_i represent the token embedding of the word i, s_i represent its sentence embedding while p_i represents its position embedding, then the embedding of a word i denoted as E_i can be represented as follows:

$$\boldsymbol{E = t_i \oplus s_i \oplus p_i} \qquad \boldsymbol{(SEQ\ Equation \setminus * ARABIC\ 1)}$$

where $\oplus$ represents the concatenation operator.

3.3 Pretrained BERT

We use the general purpose BERT model pretrained on BBC news corpus. Pretraining BERT comprises two supervised tasks. In the first task, BERT uses the concept of

masking to mask some input tokens randomly and predict the masked tokens, hence learning bidirectional representations. The hidden representations of the masked tokens are passed to the softmax layer. The second task is next sentence prediction, the purpose of which is to understand the relationship between two sentences.

3.4 Fine Tuning BERT with FARM for ADR Detection

Transfer learning represents the idea of adapting learnings from one task to another. Knowledge learned by the pretrained BERT model can be used to model any downstream task.

We use FARM to fine tune BERT for detecting ADRs from the given text sequences. FARM provides a framework that makes transfer learning with BERT simpler. It is built using transformers and provides a modular design for the language models and prediction heads. The pretrained language model is adapted to the downstream task using the prediction heads. The downstream task in our case is ADR extraction. FARM simplifies multitask learning by allowing to switch between multiple prediction heads on top of the language model. ADR detection is modelled as a sequence labelling problem in which a label is predicted for each token in the given sequence of tokens of the input text.

3.5 ADR Prediction

Given in input sequence s, weight matrix w and bias value b, the probability of the given sequence s belonging to class c is computed by the softmax function as value of the variable x

$$P(x = c|s; w; b = softmax\ (w\ .s + b)$$
$$= \frac{e^{wc.s+bc}}{\sum_{n=1}^{n}\ wn.s + bn} \quad (2)$$

where n denotes the total number of ADR categories.

3.6 Optimization

FARM-BERT is optimized using adam optimizer. The parameter update rule of adam is given as follows:

$$w_t = w_{t-1} - \eta \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \varepsilon} \quad (3)$$

where w represents weights of the model, m represents moving averages and η is the step size.

Chapter 4

Experiments and Results

In this section, we brief the experimental settings of models used for experiments. We also evaluate the models and discuss the results..

4.1 Datasets

Experiments are performed on three datasets. The first dataset is the Twitter dataset used in [11] which was created by combining two datasets i.e. Twitter ADR dataset and Attention Deficit Hyperactivity Disorder (ADHD) dataset.

Twitter ADR dataset was collected using the names of 81 drugs common in the US market [39]. The drugs used in the tweets of this dataset did not represent any specific condition but a wide range of different ADRs. The dataset was supplemented with additional ADHD dataset which contained the drug names used for treating ADHD. There are 844 tweets in the complete dataset, 95% of which contain at least 1 indication mention or ADR. The dataset is divided into 75% train data and 25% test data. Sequence labelling is usually done using the standard I-O-B scheme according to which the tokens are labelled based on their positions either at beginning (B), inside (I), or outside (O) the given entity. The Twitter data has been labelled by adopting an I-O scheme having 4 categories: I-ADR indicating the given token is a part of an ADR, I-Indication indicating the given token is a part of an indication, O-indication indicating

the token is outside any indication or ADR, and < P AD > indicating that the token is a padding.

The second dataset comprising biomedical text has been collected from PubMed abstracts [40]. There are 6,821 sentences in the dataset, each containing at least one mention of ADR. The dataset is divided into train data, validation data, and test set in the ratio of 8:1:1. The similar I-O scheme has been used for annotating the PubMed dataset. However, the dataset does not contain any I-Indication category leaving behind 3 labels for each token i.e. I-ADR, O, or < P AD >.

The third dataset is TwiMed corpus [41]. This dataset further comprises two parts, TwiMed-Twitter and TwiMed-PubMed. Three types of entities are labelled in the corpus i.e. drugs, symptoms and diseases. We consider symptoms and diseases as adverse reactions in our experiments. Moreover, there are three types of relations between these entities i.e. reason-to-use, outcome-negative, and outcome-positive. Outcome-negative indicates that drugs in the given input sequence can be a cause of adverse reactions. We consider the sentence as ADR-positive if the relationship between drugs and adverse reactions was annotated as outcome-negative. Similar considerations have also been made in the experiments conducted by Zhange et. Al [42]

4.2 Evaluation Metrics

Precision (P), Recall (R), and F Score (F) are used to evaluate the performance of the model. We choose these metrics because they have widely been used for evaluating the models in state-of-the-art works.

Precision measures relevancy of the results. In other words, it describes how many samples predicted to be belonging to a certain class actually belong to that class. It shows how often our model misclassified other classes as this class.

$$P = \frac{True\ Positives}{True\ Poistives+False\ Positives} \quad (4)$$

Recall measures how many actual relevant results have been returned. It calculates how many actual samples belonging to a certain class are correctly predicted by the model giving insight into misclassification of this class as another class.

$$R = \frac{True\ Positives}{True\ Poistives+False\ Positives} \quad (5)$$

Very often, precision and recall are inversely related to each other. To overcome this imbalance, F score is used which is the harmonic mean of precision and recall.

$$F = \frac{2.P.R}{P+R} \quad (6)$$

4.3 Proposed Model Configuration

The learning rate in FARM-BERT is set to 3e-5. The model is fine-tuned using a batch size 8 for 5 epochs.

4.4 Comparison with Baseline Models

Experiments are performed with the following conventional and deep learning models on Twitter and PubMed datasets. The results of these models are compared with the proposed model.

- Support Vector Machine (SVM): We use a linear kernel SVM to detect ADR based on word n-grams, sentence embeddings, and lexical features i.e. names of drugs and ADRs.
- Multilayer Perceptron (MLP): We use MLP classifier fed with word ngrams, sentence embeddings, and lexical features i.e. names of drugs and ADRs. Batch size is set to 16, and Adam is used as an optimizer.
- Convolutional Neural Network (CNN): We initialize the embedding layer of CNN with word embeddings. Three filters of heights 3, 4 and 5 are used in the convolutional layer.1-max pooling is applied over the convolved feature maps to select the most salient features and reduce the output dimension. The resultant features are concatenated and passed to the output layer which detects the presence of ADR in the given input sequences. We use the batch size of 16 and Adam as the optimization algorithm.
- Long Short-Term Memory (LSTM): We initialize the embedding layer of LSTM with word embeddings. The sequences returned by this layer are passed to LSTM layer followed by a dense layer. The final layer is the output layer which uses softmax

activation function to detect ADRs. We use batch size of 16 and rmsprop as an optimizer.

- Bidirectional Encoder Representations from Transformers (BERT): BERT is bidirectional transformer encoder having multiple layers. We use the pretrained BERTbase where the number of transformer blocks/ layers L is 12, hidden size H is 768, while the number of self-attention heads A is 12. The model is fine-tuned for detecting ADRs using 5 epochs. Batch size and learning are set to 16 and 2e-5 respectively.

Table 1

Models	Features	Twitter			PubMed		
		P	R	F	P	R	F
SVM	Word n-grams	0.701	0.650	0.675	0.711	0.682	0.695
	ADR Terms	0.503	0.514	0.508	0.539	0.558	0.548
	Sentence Embeddings	0.604	0.644	0.624	0.671	0.611	0.641
	Word n-grams + ADR Terms + Sentence Embeddings	0.729	0.688	0.708	0.717	0.706	0.711
MLP	Word n-grams	0.711	0.661	0.686	0.719	0.684	0.701
	ADR Terms	0.512	0.524	0.518	0.521	0.544	0.532
	Sentence Embeddings	0.615	0.645	0.630	0.685	0.666	0.675
	Word n-grams + ADR Terms + Sentence Embeddings	0.727	0.738	0.732	0.733	0.756	0.744
LSTM	Word2vec Word Embeddings	0.779	0.798	0.788	0.801	0.792	0.796
	Fasttext Word Embeddings	0.786	0.812	0.799	0.825	0.798	0.811
	Glove Word Embeddings	0.771	0.782	0.776	0.810	0.786	0.798
CNN	Word2vec Word Embeddings	0.854	0.799	0.826	0.861	0.806	0.833
	Fasttext Word Embeddings	0.863	0.801	0.832	0.877	0.819	0.848
	Glove Word Embeddings	0..843	0.803	0.823	0.872	0.798	0.835
BERT	BERT Embeddings	0.831	0.850	0.870	0.920	0.930	0.910
FARM-BERT	BERT Embeddings	0.840	0.861	**0.896**	0.982	0.964	**0.976**

Table 1: Comparison of results yielded by FARM-BERT with the results yielded by baseline models applied on Twitter and PubMed datasets

Table 1 shows the results of the baseline models and the proposed model. It is observed that deep learning techniques in general yield better results than the conventional models i.e. SVM and MLP. Among the conventional models, MLP performs better than SVM. We find that ADR and drug terms alone do not play a substantial role in identifying ADRs. This indicates that spotting keywords in the given sentence cannot lead to extracting adverse drug reactions effectively as the problem depends more on the context. Incorporating contextual information using word n-grams and semantic information using sentence embeddings improves the performance of these models. However, word n-grams in these models are represented as their term frequencies which are not enough for effective classification.

In deep learning models, words in the input sequence are represented as word embeddings, and hence, the contextual information is learned utilizing the semantic representation of the words in the form of embeddings through multiple layers of the network. Among deep neural networks, BERT performs better than CNN, and CNN performs better than LSTM. We find CNN performing better than LSTM because CNNs capture local patterns while LSTMs capture global patterns in the input. We observe that in most of the cases, input sequences comprise short text. Hence, information from the local key phrases which is effectively extracted by applying CNN plays a primary role in ADR extraction. LSTMs on the other hand are good at capturing long

range dependencies. When applying LSTM, the input sentence is encoded as a long example. As a result, some important phrases may not be learned as a salient feature. We also observe the effects of different embedding models i.e. word2vec, Fasttext and Glove on CNN and LSTM. We find that both CNN and LSTM perform better when initialized with Fasttext embeddings than word2vec and glove embeddings. Fasttext model takes into account morphology of the words by extracting information from the internal structure of the words rather than considering just the whole words in the context. Fasttext represents each word by the sum of their char n-grams. By considering the subword information, fasttext, unlike word2vec and glove, generates the embeddings for out of vocabulary words as well. The training data used for any machine learning model, no matter how big it may be, can still not include all the words in a language's vocabulary. If such unseen words are found in the test data, their representations are not generated by word2vec and glove embedding models. However, fasttext overcomes this limitation and represents the out of vocabulary words by adding the embeddings for the constituent char n-grams found in the vocabulary.

BERT outperforms both CNN and LSTM. The reason for better performance of BERT is that it learns contextualized embeddings in bidirectional way. In natural language, a word is likely to convey multiple meanings based on the context in which it is used. Word2vec, fasttext and glove produce the same representations of a word even if it has different meanings in different contexts. BERT, on the other hand, produces

context dependent embeddings of a word. In BERT, an input word is represented by the sum of its token embeddings, sentence embeddings and position embeddings.

The proposed model FARM-BERT outperforms all the models by yielding the F-Scores of 89.6% and 97.6% on Twitter and PubMed datasets respectively. FARM-BERT performs better than BERT by 2% on Twitter and by 6% on PubMed datasets. Better performance of FARM-BERT than the standard BERT indicates the effectiveness of fine-tuning BERT with FARM with the modified values of hyperparameters.

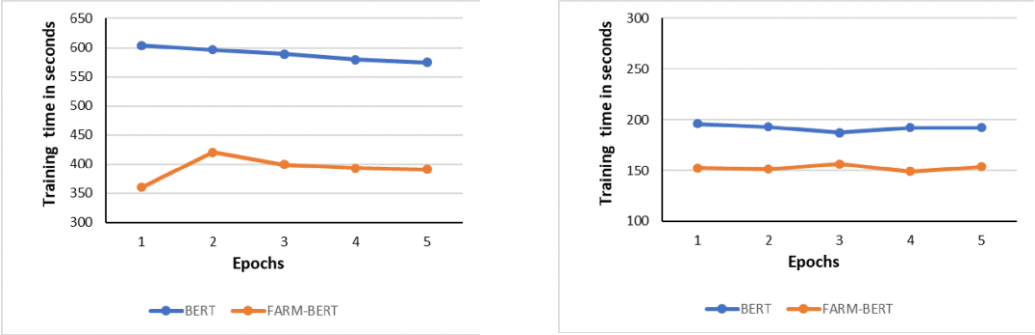
4.5 Comparison of computational performance of FARM-BERT with BERT

In this section, we compare the computational time consumed by training and testing BERT and FARM-BERT on Twitter and PubMed datasets. Table 2 shows the computation time of training both the models for each epoch in seconds while Table 3 shows the test time of both the models in seconds. Training time of both the models on PubMed and Twitter datasets is also demonstrated in Figure 2a and Figure 2b respectively. Similarly, test time of both the models on both the datasets is demonstrated in Figure 3.

Table 2

Epochs	Training time of the models			
	PubMed Dataset		Twitter Dataset	
	BERT	FARM-BERT	BERT	FARM-BERT
Epoch 1	604.01	**360.37**	192.21	**152.44**
Epoch 2	596.36	**420.06**	193.04	**151.31**
Epoch 3	589.01	**399.42**	187.22	**156.51**
Epoch 4	579.32	**393.21**	192.11	**149.22**
Epoch 5	574.50	**391.40**	192.38	**153.49**

Table 2: Comparison of training time of BERT and FARM-BERT for each epoch on Twitter and PubMed datasets



(a) Training time of BERT and FARM-BERT for each epoch on PubMed dataset

(b) Training time of BERT and FARM BERT for each epoch on Twitter dataset

Figure 2

Figure 2: Training time of BERT and FARM-BERT for each epoch on Twitter and PubMed datasets

Table 3

Test time of the models		
	PubMed Dataset	**Twitter Dataset**
BERT	46.1	22.51
FARM-BERT	**33.4**	**18.32**

Table 3: Comparison of test time of BERT and FARM-BERT on Twitter and PubMed datasets

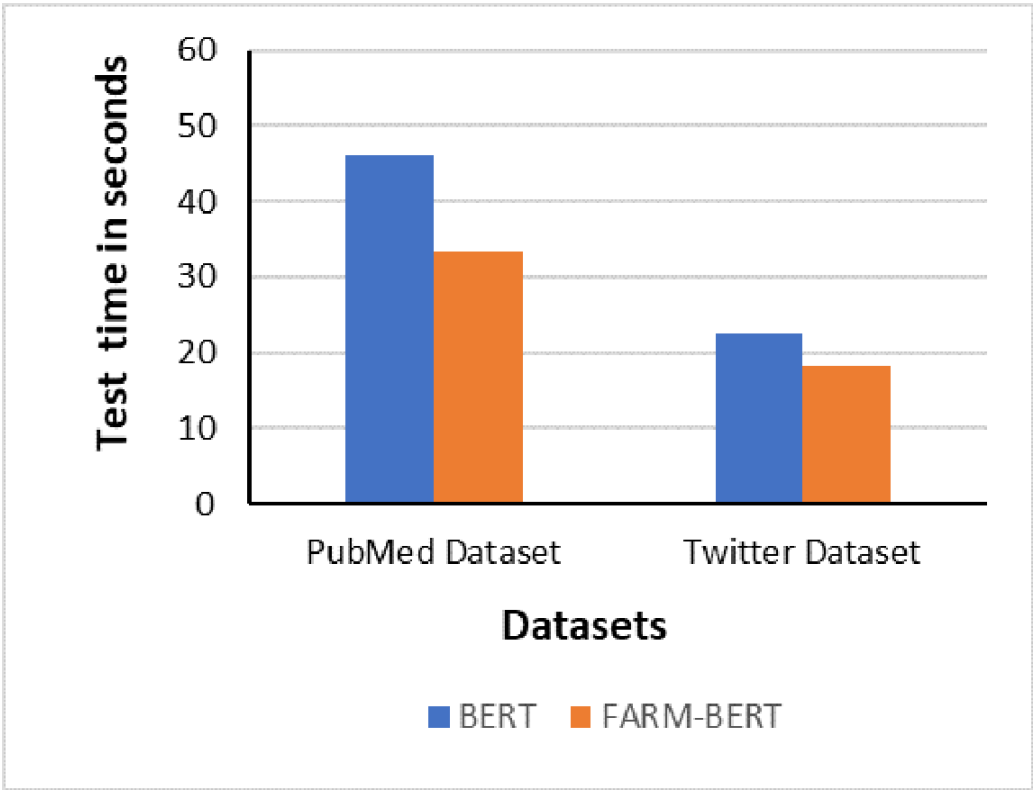


Figure 3

Figure 3: Test time of BERT and FARM-BERT on Twitter and PubMed datasets

The experiments show that training BERT in each epoch takes more time than training FARM-BERT. Similar observations have been made while testing BERT and FARM-BERT. Hence, FARM-BERT works computationally faster than the standard BERT during both training and testing. FARM-BERT is computationally faster than BERT because FARM supports parallel processing. Furthermore, support for using multiple prediction heads for multi-task learning also makes FARM-BERT faster than the standard BERT. The analysis of the computational performance of both of the models indicate the effectiveness of using FARM-BERT for ADR prediction instead of the standard BERT.

4.6 Comparison with State-of-the-Art Works

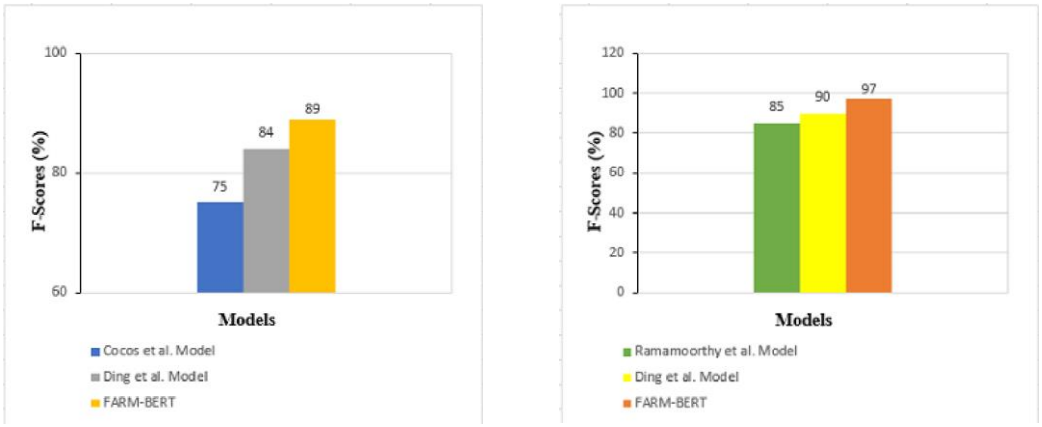
In this section we compare the results of our proposed approach with the state-of-the-art works performed on the three datasets i.e PubMed dataset, Twitter dataset, and TwiMed dataset.

Table 4 tabulates the results of the proposed method and previous works performed on PubMed and Twitter datasets. F-Scores achieved by these models are visually displayed in Figure 4a and Figure 4b respectively.

Table 4

Models	Twitter			PubMed		
	P	R	F	P	R	F
Cocos et al. [11]	0.704	0.829	0.755	_	_	_
Ramamoorthy et al. [8]	_	_	_	0.884	0.824	0.853
Ding et al. [10]	0.785	0.914	0.844	0.867	0.948	0.906
FARM-BERT	**0.84**	**0.861**	**0.896**	**0.982**	**0.964**	**0.976**

Table 4: Comparison of results yielded by FARM-BERT with the results yielded by state-of-the-art models on Twitter and PubMed datasets



(a) F-Scores yielded on Twitter dataset

(b) F-Scores yielded on PubMed dataset

Figure 4

Figure 4: F-Scores achieved by different models on Twitter and PubMed datasets

The comparisons are made with the works performed by Cocos et al. [11], Ramamoorthy et al. [8], and Ding et al. [10]. The model by Cocos et al. [11] uses BLSTM which combines forward and reverse RNNs. 400 dimensional pretrained embeddings are used to initialize the embedding layer [43]. The model has been applied on Twitter dataset. Ramamoorthy et al. [8] uses BLSTM initialized with a combination of charCNN embedding, word2vec word embedding and PoS embeddings. The model uses a self-attention mechanism and has been applied on a PubMed dataset. Ding et al. [10] uses BGRU with a combination of charLSTM embeddings and 300-dimensional Glove word

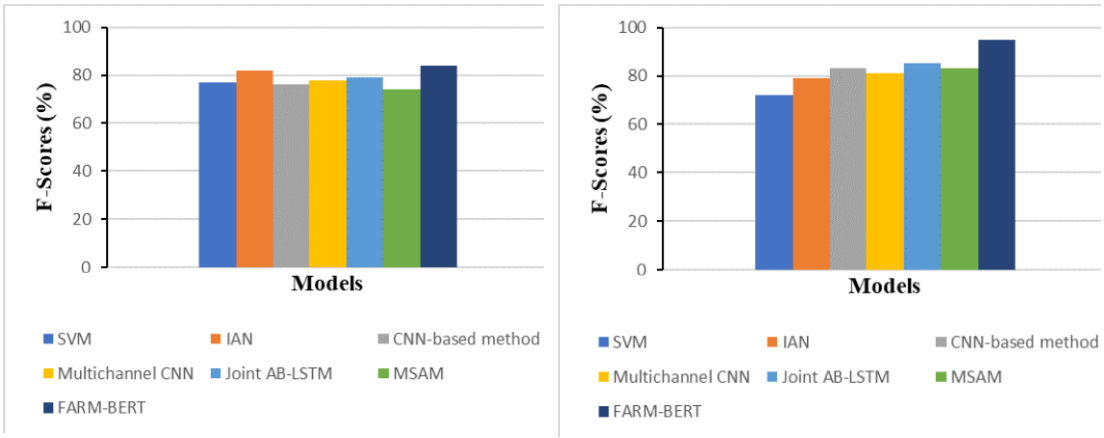
representations [43] through embedding level attention mechanism. The output of the embedding level attention layer is used as an auxiliary classifier and added to the BGRU output layer to identify ADRs. This model has been applied on both PubMed and Twitter datasets. It is evident from table 4 that the proposed model FARM-BERT outperforms all the state-of-the-art models applied on Twitter and PubMed datasets. In terms of F-score, FARM-BERT performs better than Cocos et al. [11] by approximately 14% on Twitter dataset. It performs better than Ramamoorthy et al. [8] by approximately 10% on PubMed dataset. It yields better performance than Ding et al. [10] by approximately 5% and 7% on Twitter and PubMed datasets respectively.

Table 5 compares the results achieved by FARM-BERT with results achieved by the previous works on TwiMed corpus. F-scores of the models on Twimed-Twitter and TwiMed-PubMed datasets are also demonstrated in Figure 5a and Figure 5b respectively.

Table 5

Models	TwiMed-Twitter			TwiMed-PubMed		
	P	R	F	P	R	F
SVM [44]	0.752	0.810	0.778	0.799	0.681	0.728
IAN [44]	0.836	0.813	0.824	0.878	0.738	0.792
CNN-based method [45]	0.739	0.788	0.761	0.849	0.831	0.835
Multichannel CNN [46]	0.738	0.841	0.780	0.861	0.780	0.816
Joint AB-LSTM [47]	0.748	0.856	0.799	0.858	0.852	0.853
MSAM [42]	0.701	0.828	0.754	0.817	0.856	0.831
FARM-BERT	**0.831**	**0.868**	**0.849**	**0.952**	**0.966**	**0.959**

Table 5: Comparison of results yielded by FARM-BERT with the results yielded by state-of-the-art models on Twimed corpus



(a) F-Scores yielded on TwiMed-Twitter dataset

(b) F-Scores yielded on TwiMed-PubMed dataset

Figure 5

Figure 5: F-Scores yielded by different models on TwiMed dataset

The first two models in Table 5 i.e. SVM and interactive attention network (IAN) have been used by Alimova et al. [44] on TwiMed dataset. IAN uses attention mechanism to

learn target and contextual representations. The experiments using CNN based method, multichannel CNN, joint AB-LSTM, and Multihop Self-Attention Mechanism (MSAM) have been performed by [42] on TwiMed corpus. CNN based method was proposed by Liu et al. [45] and Quan et al. [46] for relationship detection. Joint AB-LSTM was proposed by Kumar et al. [47]. MSAM has been proposed by [27] which uses a multihop mechanism to learn complex semantic information by focusing on different segments of a sentence. It can be seen from the table that the FARM-BERT approach proposed by our work performs better than all the other approaches.

Chapter 5

Conclusion

In this work we tuned BERT with FARM using multi-task learning to present an end-to-end solution for identifying ADRs on Twitter, PubMed and TwiMed datasets. The proposed model FARM-BERT uses modified hyperparameters as compared to the standard BERT. These hyperparameters include different learning rate, number of epochs, and batch size. We performed multiple experiments and compared the results with different baseline models i.e. SVM, MLP, CNN, LSTM, and standard BERT. We also compared the results with other state-of-the-art works. Experiments show that the proposed FARM-BERT outperforms all the models yielding the F-scores of 89.6%, 97.6%, 84.9%, and 95.9% on Twitter, PubMed, TwiMed-Twitter and TwiMed-PubMed datasets respectively.

Chapter 6

References

[1] W. H. Organization, et al., International drug monitoring: the role of national centres, report of a WHO meeting [held in Geneva from 20 to 25 September 1971], World Health Organization, 1972.

[2] J. C. Bouvy, M. L. De Bruin, M. A. Koopmanschap, Epidemiology of adverse drug reactions in europe: a review of recent observational studies, Drug safety 38 (5) (2015) 437-453.

[3] E. Commission, Proposal for a regulation amending, as regards pharmacovigilance of medicinal products for human use. regulation (ec) no 726/2004. impact assessment.

[4] S. R. Ahmad, Adverse drug event monitoring at the food and drug administration, Journal of general internal medicine 18 (1) (2003) 57-60.

[5] C. C. Freifeld, J. S. Brownstein, C. M. Menone, W. Bao, R. Filice, T. Kass Hout, N. Dasgupta, Digital drug safety surveillance: monitoring pharmaceutical products in twitter, Drug safety 37 (5) (2014) 343-350.

[6] S. T. Aroyehun, A. Gelbukh, Detection of adverse drug reaction in tweets using a combination of heterogeneous word embeddings, in: Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task, 2019, pp. 133-135.

[7] A. Breden, L. Moore, Detecting adverse drug reactions from twitter through domain-specific preprocessing and bert ensembling, arXiv preprintarXiv:2005.06634.

[8] S. Ramamoorthy, S. Murugan, An attentive sequence model for adverse drug event extraction from biomedical text, arXiv preprintarXiv:1801.00625.

[9] F. Li, Y. Zhang, M. Zhang, D. Ji, Joint models for extracting adverse drug events from biomedical text., in: IJCAI, Vol. 2016, 2016, pp. 2838-2844.

[10] P. Ding, X. Zhou, X. Zhang, J. Wang, Z. Lei, An attentive neural sequence labeling model for adverse drug reactions mentions extraction, IEEE Access 6 (2018) 73305-73315.

[11] A. Cocos, A. G. Fiks, A. J. Masino, Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts, Journal of the American Medical Informatics Association 24 (4) (2017) 813-821.

[12] X. Liu, H. Chen, Azdrugminer: an information extraction system for mining patient-reported adverse drug events in online patient forums, in: International conference on smart health, Springer, 2013, pp. 134-150.

[13] M. Rastegar-Mojarad, R. K. Elayavilli, Y. Yu, H. Liu, Detecting signals in noisy data-can ensemble classifiers help identify adverse drug reaction in tweets, in: Proceedings

of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing, 2016.

[14] M. Z. Sh, T. EV, T. AE, Identifying disease-related expressions in reviews using conditional random fields, Computational Linguistics and Intellectual Technologies (2017) 155-166.

[15] H.-J. Dai, M. Touray, J. Jonnagaddala, S. Syed-Abdul, Feature engineering for recognizing adverse drug reactions from twitter posts, Information 7 (2) (2016) 27.

[16] D. S. Miranda, Automated detection of adverse drug reactions in the biomedical literature using convolutional neural networks and biomedical word embeddings, arXiv preprint arXiv:1804.09148.

[17] Z. Li, H. Lin, W. Zheng, An effective emotional expression and knowledge-enhanced method for detecting adverse drug reactions, IEEE Access 8 (2020) 87083-87093.

[18] I. Alimova, E. Tutubalina, Automated detection of adverse drug reactions from social media posts with machine learning, in: International Conference on Analysis of Images, Social Networks and Texts, Springer, 2017, pp. 3-15.

[19] A. Sarker, G. Gonzalez, Portable automatic text classification for adverse drug reaction detection via multi-corpus training, Journal of biomedical informatics 53 (2015) 196-207.

[20] J. Bian, U. Topaloglu, F. Yu, Towards large-scale twitter mining for drug related adverse events, in: Proceedings of the 2012 international workshop on Smart health and wellbeing, 2012, pp. 25-32.

[21] S. Kiritchenko, S. M. Mohammad, J. Morin, B. de Bruijn, Nrc-canada at smm4h shared task: classifying tweets mentioning adverse drug reactions and medication intake, arXiv preprint arXiv:1805.04558.

[22] E. Aramaki, Y. Miura, M. Tonoike, T. Ohkuma, H. Masuichi, K. Waki, K. Ohe, Extraction of adverse drug effects from clinical records., MedInfo 160 (2010) 739-743.

[23] S. Moen, T. S. S. Ananiadou, Distributional semantics resources for biomedical text processing, Proceedings of LBM (2013) 39-44.

[24] S. Chowdhury, C. Zhang, P. S. Yu, Multi-task pharmacovigilance mining from social media posts, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 117-126.

[25] S. Gupta, M. Gupta, V. Varma, S. Pawar, N. Ramrakhiyani, G. K. Palshikar, Co-training for extraction of adverse drug reaction mentions from tweets, in: European Conference on Information Retrieval, Springer, 2018, pp. 556-562.

[26] S. Gupta, S. Pawar, N. Ramrakhiyani, G. K. Palshikar, V. Varma, Semisupervised recurrent neural network for adverse drug reaction mention extraction, BMC bioinformatics 19 (8) (2018) 212.

[27] M. Zhang, G. Geng, Adverse drug event detection using a weakly supervised convolutional neural network and recurrent neural network model, Information 10 (9) (2019) 276.

[28] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991.

[29] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360.

[30] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv preprint arXiv:1603.01354.

[31] E. Tutubalina, S. Nikolenko, Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews, Journal of healthcare engineering 2017.

[32] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[33] D. Weissenbacher, A. Sarker, A. Magge, A. Daughton, K. O'Connor, M. Paul, G. Gonzalez, Overview of the fourth social media mining for health (smm4h) shared tasks at acl 2019, in: Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task, 2019, pp. 21-30.

[34] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.

[35] S. Chen, Y. Huang, X. Huang, H. Qin, J. Yan, B. Tang, Hitsz-icrc: A report for smm4h shared task 2019-automatic classification and extraction of adverse drug reactions in tweets, ACL 2019 (2019) 47.

[36] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234-1240.

[37] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, M. McDermott, Publicly available clinical bert embeddings, arXiv preprint arXiv:1904.03323.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998-6008.

[39] A. Nikfarjam, A. Sarker, K. O'connor, R. Ginn, G. Gonzalez, Pharma-

covigilance from social media: mining adverse drug reaction mentions us-

ing sequence labeling with word embedding cluster features, Journal of the

American Medical Informatics Association 22 (3) (2015) 671-681.

[40] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann- Apitius, L. Toldo, Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports, Journal of biomedical informatics 45 (5) (2012) 885-892.

[41] N. Alvaro, Y. Miyao, N. Collier, Twimed: Twitter and pubmed comparable corpus of drugs, diseases, symptoms, and their relations, JMIR public health and surveillance 3 (2) (2017) e24.

[42] T. Zhang, H. Lin, Y. Ren, L. Yang, B. Xu, Z. Yang, J. Wang, Y. Zhang, Adverse drug reaction detection via a multihop self-attention mechanism, BMC bioinformatics 20 (1) (2019) 479.

[43] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532-1543.

[44] I. Alimova, V. Solovyev, Interactive attention network for adverse drug reaction classification, in: Conference on Artificial Intelligence and Natural Language, Springer, 2018, pp. 185-196.

[45] S. Liu, B. Tang, Q. Chen, X. Wang, Drug-drug interaction extraction via convolutional neural networks, Computational and mathematical methods in medicine 2016.

[46] C. Quan, L. Hua, X. Sun, W. Bai, Multichannel convolutional neural network for biological relation extraction, BioMed research international 2016.

[47] S. K. Sahu, A. Anand, Drug-drug interaction extraction from biomedical texts using long short-term memory network, Journal of biomedical informatics 86 (2018) 15-24.