A Unified Multi-task Adversarial Learning Framework for Pharmacovigilance Mining

Shweta Yadav, Asif Ekbal, Sriparna Saha, Pushpak Bhattacharyya
Indian Institute of Technology Patna
Patna, India
{shweta.pcs14,asif,sriparna,pb}@iitp.ac.in

Abstract

The mining of adverse drug reaction (ADR) has a crucial role in the pharmacovigilance. The traditional ways of identifying ADR are reliable but time-consuming, non-scalable and offer a very limited amount of ADR relevant information. With the unprecedented growth of information sources in the forms of social media texts (Twitter, Blogs, Reviews etc.), biomedical literature, and Electronic Medical Records (EMR), it has become crucial to extract the most pertinent ADR related information from these free-form texts. In this paper, we propose a neural network inspired multi-task learning framework that can simultaneously extract ADRs from various sources. We adopt a novel adversarial learning-based approach to learn features across multiple ADR information sources. Unlike the other existing techniques, our approach is capable to extracting fine-grained information (such as *'Indications'*, *'Symptoms'*, *'Finding'*, *'Disease'*, *'Drug'*) which provide important cues in pharmacovigilance. We evaluate our proposed approach on three publicly available real-world benchmark pharmacovigilance datasets, a Twitter dataset from PSB 2016 Social Media Shared Task, CADEC corpus and Medline ADR corpus. Experiments show that our unified framework achieves state-of-the-art performance on individual tasks associated with the different benchmark datasets. This establishes the fact that our proposed approach is generic, which enables it to achieve high performance on the diverse datasets. The source code is available here[1].

1 Introduction

Early detection and monitoring of adverse drug reactions (ADRs) can minimize the deleterious impact on patients and health-care systems (Hakkarainen et al., 2012; Sultana et al., 2013). For prevention, the drug safety organizations known as pharmacovigilance agencies conduct post-market surveillance to identify the drug's side effects post-release. However, the majority of the existing ADE surveillance systems utilizes passive spontaneous reporting system databases, such as the Federal Drug Administration's Adverse Event Reporting System (FAERS) (Li et al., 2014). These systems are often under-reported, biased and delayed. To overcome the limitation of a passive reporting system, active methods to ADR monitoring continuously explores frequently updated ADR data sources (Behrman et al., 2011).

The quantity and near-instantaneous nature of social media provide potential opportunities for real-time monitoring of Adverse Drug Reaction (ADR). The fact that this data is up-to-date and is generated by patients overcomes the weaknesses of traditional ADR surveillance techniques (Leaman et al., 2010). Thus, social media could complement traditional information sources for more effective pharmacovigilance studies, as well as potentially serve as an early warning system for unknown ADR, which may be important for a clinical decision. Additionally, the high statistically significant correlation ($p < 0.001$, $\rho = 0.75$) between FAERS and ADRs (extracted through Twitter data) shows that Twitter is a viable pharmacovigilance data source (Freifeld et al., 2014).

With the enormous amount of data generated every day, it is desirable to have an automated ADR extraction system that can ease the work of domain experts to quickly investigate the vast amount of unstructured text and identify emerging trends. This may correspond to mapping previously undiscovered adverse effect with a given drug, or discovering an unforeseen impact to a change in the manufacturing process. However, extracting this information from the unstructured text poses several challenges as follows:

[1] https://bit.ly/2EMln36

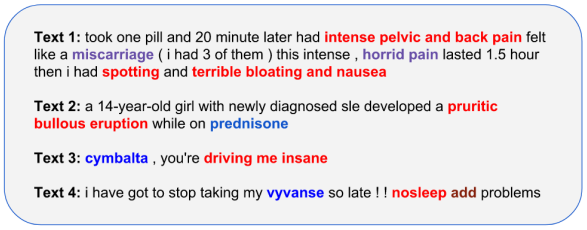

Figure 1: Sample sentences from CADEC (Text1), MEDLINE (Text 2) and Twitter (Text 3,4) dataset. The token in red represents *ADR*, purple denotes *Finding*, blue represent *Drug name* and brown colour text represents *Indication*.

- **Multiple Context:** Context carries an essential role in determining the semantic labels of the medical concepts. For example, consider the following tweets:
  Tweet 1: "*Advil cured my horrific pain, but made my stomach upset*"
  Tweet 2: "*Advil cured my upset stomach but gave me a horrific pain*"
  The above tweets, although have a similar medical concept, their contexts specify the associated class types. In Tweet 1, 'pain' refers to the class type Symptom, while in Tweet 2, it refers to ADR.

- **Multiple word form:** Social media text offers some inherently distinct challenges such as containing short word-forms (eg,"*need to sleep 24/7*"), misspelled wordforms (eg, "*fluoxetine, it just make me so tiered* '), abbreviated words (eg, CIT for *Citopram*), slangs (eg, "seroquel knocked me out"), implicit sense (eg, "*hard time getting some Z's*"), symbols (such as emoticons), and figurative languages (eg, "*quetiapine zombie*"). This arbitrariness increases the difficulty level in capturing the semantic relationships between the different types.

To overcome these limitations, several machine learning and deep learning models are introduced for ADR mining. However, these models are very task-specific and often fail to show reasonable accuracies when these evaluated for some other domains or other annotation schemes.

In this paper, we propose a unified multi-task learning (MTL) framework that works on the concept of adversarial learning. Our model is capable of learning several tasks associated with ADR monitoring with different levels of supervisions collectively. The proposed approach differs from the previous studies in two aspects:
Firstly, most of the existing methods in multi-task learning attempt to divide the features of different tasks based on task-specific and task-invariant feature space, considering only component-wise parameters. The major drawback of this mechanism is that the common feature space often incorporates the task-specific feature space, leading to feature redundancy. Given this issue in multi-task learning (MTL), in our proposed framework we employ adversarial learning (Goodfellow et al., 2014), which helps in eliminating redundant features from the feature space and prevent the contamination between shared and task-specific features. **Secondly**, we also employ the highway and residual connection whenever necessary to avoid the vanishing gradient problem and improve the performance of our deep neural model (multi-headed attention based stacked recurrent and convolutional neural network).
Contributions:
Contributions of our current work can be summarized as follows:
(1) We propose a unified multi-task learning (MTL) framework for pharmacovigilance mining that exploits the capabilities of adversarial learning to learn the shared complementary features across the multiple ADR datasets. To our best knowledge, this is the very first attempt to study the effect of adversarial learning method in MTL environment, especially for pharmacovigilance mining.
(2) Our proposed model is capable of automatically identifying the various information (such as *Symptom*, *Finding*, *Disease*, *Drug*), in addition to the ADR.
(3) We validate our proposed framework on three popular benchmark datasets, namely Twitter (Sarker et al., 2016), CADEC (Karimi et al., 2015) and MEDLINE (Gurulingappa et al., 2012a) for pharmacovigilance mining, having different annotation schemes. We extract the following tags: *ADR*, *Drugs*, and *Indications* from the Twitter dataset, *ADR*, *Disease*, *Drug*, *Finding*; and *Symptom* from the CADEC dataset; and *Drug* and *ADR* mentions from the MEDLINE dataset. Figure-1 shows exemplary sentences from each dataset.
(4) Our unified multi-task model achieves the state-of-the-art performance in the ADR labeling and outperforms the strong baseline models for all the other pharmacovigilance labels.

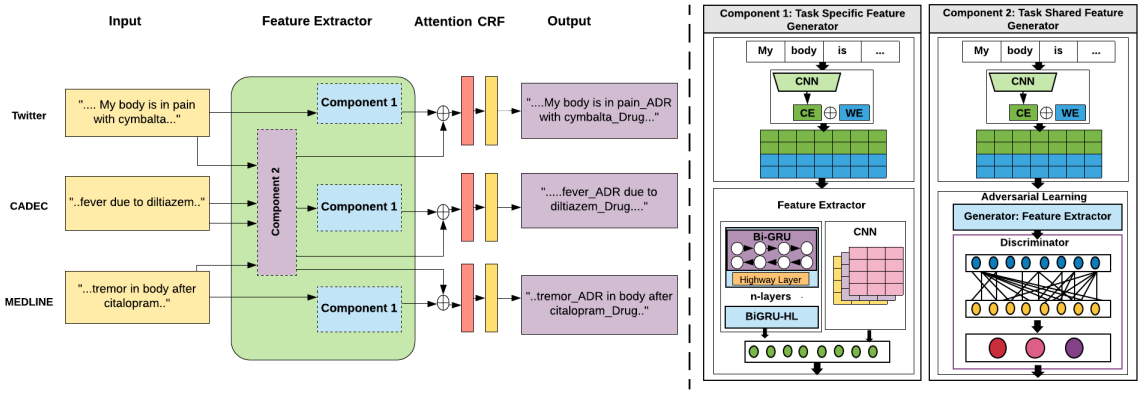

Figure 2: Proposed model architecture for pharmacovigilance mining. (all the neurons representation are hypothetical). The right part of the image describes the *Component 1* and *Component 2*.

2 Related Work

Depending upon the source of data, we categorize the previous works as:

(i) Biomedical Text and Electronic Medical Record:

Several Natural Language Processing (NLP) techniques have been proposed to extract ADRs from the Electronic Medical Record (Wang et al., 2009; Friedman, 2009; Aramaki et al., 2010) and medical case reports (Gurulingappa et al., 2011). Gurulingappa et al. (2012a) adapted machine learning technique for the identification and extraction of potential adverse drug event relations from the MEDLINE case reports. Unlike other spontaneous data sources such as social media, both EMR and medical case reports offer several advantages of having complete records of patients' medical history, treatment, conditions and the possible risk factors, and is also not restricted to the patients experiencing ADRs (Harpaz et al., 2012b). Recently, a study conducted by (Sarker and Gonzalez, 2015) utilized the data from MEDLINE case reports and Twitter. They proposed several textual features and investigated how the combination of different datasets would increase the performance of identifying ADRs. With the advancement of the neural network technique, (Huynh et al., 2016) investigated multiple neural network (NN) frameworks for ADR classification on both medical case reports and Twitter dataset.

(ii) Social Media: Social media offers a very rich and viable source of information for identifying potential ADRs in a real-time. Leaman et al. (2010) conducted very first study utilizing user comments from their social media post. In total, the dataset contains $6,890$ user comments. The research shows that user comments are highly beneficial in uncovering the ADRs. Further works (Gurulingappa et al., 2012b; Benton et al., 2011; Harpaz et al., 2012a) utilized the lexicon-based approach to extract the ADRs. However, these approaches are only restricted to a number of target ADRs. Nikfarjam and Gonzalez (2011) exploited rule-based technique over naive lexicon-based approach on the same dataset which was capable of detecting ADR not included in lexicons.

With the emergence of annotated data, several research works have employed supervised machine learning techniques such as Support Vector Machine (SVM) (Sarker and Gonzalez, 2015), Conditional Random Field (CRF) (Nikfarjam et al., 2015) and Random Forest (Zhang et al., 2016).

In recent years with the introduction of deep learning techniques, most of the studies utilize deep learning model to predict ADRs. Lee et al. (2017) developed semi-supervised deep learning model on the Twitter corpus. In particular, they used the Convolution Neural Network (CNN) for classification. Stanovsky et al. (2017) used the Recurrent Neural Network integrated with knowledge graph embedding on the CADEC corpus. Their study shows that this integration can make the model more accurate. Tutubalina and Nikolenko (2017) explored the combination of CRF and Recurrent Neural Network (RNN). Their

results show that CRF can assist RNN model in capturing the context well. The most relevant work to this study is the work conducted by Chowdhury et al. (2018). They learned jointly for three tasks: binary classification, ADR labeling, and indication labeling using RNN-attention-coverage model.

3 Methodology

With our adversarial multi-task framework, we jointly learn to label the ADR events from multiple ADR datasets. ADR labeling is a sequence labeling problem. For a given input sequence X, the model learns to find the optimal tag sequence y^*. Mathematically,

$$y^* = \arg\max_y P(Y|X) \quad (1)$$

Our proposed adversarial multi-task framework is depicted in Figure 2. Our model comprises of five components:

(1) Embedding Layer: It captures the meaning and semantic associations between pharmacovigilance word that appears in the text.

(2) Encoder/Feature Extractor Layer, which generates both task-specific and task-shared feature. Each of these feature generator modules consists of Convolutional Neural Network (CNN) followed by stacked Bi-Gated Recurrent Unit (GRU). Task-specific feature generator is responsible for capturing the features specific to the task. In the task-shared feature generator, there is an additional adversarial learning component, where feature extractor (Generator) is working operates adversarially towards a learnable multi-layer perceptron (Discriminator), preventing it from making an accurate prediction about the types of the task the feature generated from.

(3) Concatenation Layer: This is responsible for concatenating the feature representation obtained by both the feature extractor modules.

(4) Multi-head Attention Layer: This learns to encode better the given word by looking at the other words in the text.

(5) CRF Layer: This is used to predict the most probable tag sequence.

3.1 Input Text

The input to our model is a sequence of words $X = (x_1, x_2, \ldots, x_n)$ corresponding to social-media posts/medical case reports comprising of n words.

3.2 Embedding Layer

This layer generates two forms of representations:
Word embedding: maps each word x_i to low dimensional vector $w_i \in \mathbb{R}^{d_e}$. We use pre-trained word embedding of dimension d_e.
Character embedding:: to capture the morphological features. The character embedding can help in capturing the representations of the out of vocabulary (OOV) words, misspelt words and variations in noun or verb phrase. When it comes to the social media text, this issue even becomes more crucial to resolve. Character embedding is one of the ways to resolve this issue. It allows the model to learn lexical patterns (e.g. suffix or prefix) which eventually helps in capturing the out-of-vocabulary (OOV) words and some other information which is difficult to capture through word embedding.

We employ CNN for character embedding.

Let $C = \{c_1, c_2, \ldots, c_k\}$ be the character sequence of words x_i having length l. Each character c_j is represented as a one-hot vector of length C, which is the number of unique characters in the dataset. The resulted one-hot representations of all the characters in the word are stacked to form a matrix $M \in \mathbb{R}^{k \times |C|}$. Thereafter, we apply several filters of different width to this matrix. The width of these filters varies from 1 to k, i.e., these filters look at 1 to k-gram character sequences. The max-pooling operation is performed followed by the convolutional operation to pick the most relevant feature. We call this character embedding feature as c_i.

Finally, the output of word embedding for the i^{th} word is the concatenation of word embedding w_i and the character embedding c_i. For each $x_i \in X$, the embedding layer generate the embedding in the following way:

$$e_i = w_i \oplus c_i \quad (2)$$

3.3 Feature Extractor

Our feature extractor utilizes CNN and stacked Bi-GRU to encode the output of the *Embedding layer*. CNN and stacked Bi-GRU takes the Embedding layer output as input and generate the features to further encode the sequence information. Since, we employ the stacked Bi-GRU, there could be vanishing gradient problem. To tackle this, we employ highway layer (Srivastava et al., 2015), that has shown a significant impact in reducing vanishing gradient problem in various NLP tasks (Kim

et al., 2016; Costa-jussà and Fonollosa, 2016).

Let us the consider the input sequence to this layer is $E = \{e_1, e_2, \ldots, e_n\}$. A convolution operation is performed over the zero-padded sequence E^p. Similar to the character embedding, a set of k filter of size m are applied to the sequence. We obtain convoluted features c_t at given time t for $t = 1, 2, \ldots, n$.

$$c_t = relu(F[e_{t-\frac{m-1}{2}} \ldots e_t \ldots e_{t+\frac{m-1}{2}}]) \quad (3)$$

Then, we generate the feature vectors $C' = [c'_1, c'_2 \ldots c'_n]$, by applying max pooling on C.

Inspired by the success of stacked attentive RNN in solving other NLP tasks (Wu et al., 2016; Graves et al., 2013; Dyer et al., 2015; Prakash et al., 2016), we use the stacked GRU to encode the input text. The stacked GRU is an extension to GRU model that has multiple hidden GRU layers. The purpose of using multiple GRUs layers is to learn more sophisticated conditional distributions from the data (Bahdanau et al., 2015). In this work, we employ vertical stacking strategy where the output of the previous layer of GRU is fed to the highway layer and corresponding output is passed as input to the next layer of GRU. Let the number of layers in stacked GRU is L then the GRU computes the hidden state for each layer $l \in L$ as follows:

$$h_k^l = GRU(h_k^{l-1}, h_{k-1}^l) \quad (4)$$

where, h_k^l is the hidden state representation at l^{th} layer. The input h_k^0 to the first layer ($l = 1$) of GRU are initialized randomly. The first layer of GRU unit at k^{th} word feature takes the input as the embedding layer output e_k of the k^{th} word. We compute the forward ($\overrightarrow{h_k}$) and backward ($\overleftarrow{h_k}$) hidden state for each word k in the sentence. The final hidden state at layer $l \in L$ is computed by augmenting both the hidden states: $z_k^l = [\overrightarrow{h_k^l} \oplus \overleftarrow{h_k^l}]$. The final input text representation from stacked Bi-GRU layer is calculated by taking the hidden state of the last layer (L) of the GRU as follows:

$$h_1, h_2, \ldots, h_n = [\overrightarrow{h_1^L} \oplus \overleftarrow{h_1^L}], [\overrightarrow{h_2^L} \oplus \overleftarrow{h_2^L}], \ldots, [\overrightarrow{h_n^L} \oplus \overleftarrow{h_n^L}] \quad (5)$$

We compute the overall input text representation by concatenating the output of CNN layer C' and stacked Bi-GRU (eq. 5) as follows:

$$z_1, z_2, \ldots, z_n = [c'_1 \oplus h_1], [c'_2 \oplus h_2], \ldots, [c'_n \oplus h_n] \quad (6)$$

The above approach to generate task specific feature is computed at for each task separately. In order to capture the common features along the task, we utilize the above feature extractor framework which serves as a Generator model and the feed forward neural network as a Discriminator.

3.4 Task Discriminator Layer

Our feature extractor layer is generating two types of features, shared and task-specific. Ideally both feature spaces should be mutually exclusive. To ensure that task-specific features of given task do not exist in the shared space, we exploit the concept of adversarial training (Goodfellow et al., 2014) into shared feature space. We follow the same method as introduced by (Liu et al., 2017) to make the shared feature space uncontaminated by the task-specific features.

For achieving the aforementioned strategy, a `Task Discriminator` D is used to map the attention prioritized shared feature to estimate the task of its origin. In our case, `Task Discriminator` is a fully connected layer using a softmax layer to produce the probability distribution of the shared features belonging to any task. The shared feature extractor (c.f. 3.3) works as `Generator` (G) to generate shared features. The shared feature extractor is made to work in an adversarial way, preventing the discriminator from predicting the task and hence preventing contamination in the shared space. The adversarial loss is used to train the model. Let us assume that the shared feature (c.f. equation 6) is $\{z_1^s, z_2^s, \ldots, z_n^s\}$. It can be represented as:

$$D(z^s) = softmax(z_n^s W^d + b^d) \quad (7)$$

where W^d and b^d are the weight matrix and bias, respectively.

3.5 Concatenation Layer

Let us denote the shared and task-specific features for input text are $z^s = \{z_1^s, z_2^s, \ldots, z_n^s\}$ and $z^t = \{z_1^t, z_2^t, \ldots, z_n^t\}$. Finally, the output of the feature extractor layer is computed as the concatenation of the shared and task-specific feature as follows:

$$\begin{aligned} S &= z_1^s \oplus z_1^t, z_2^s \oplus z_2^t, \ldots, z_n^s \oplus z_n^t \\ &= s_1, s_2, \ldots, s_{n-1}, s_n \end{aligned} \quad (8)$$

3.6 Multi-head Attention Layer

The multi-head attention is used to learn the dependencies between any pair of words in the input text. We apply the multi-head attention on the final representation of the input text S as computed in Equation 8. The multi-head attention (Vaswani

et al., 2017) can be precisely described as follows:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V \quad (9)$$

where, Q, K and V are the query, key and value matrix. In our experiment, all these values are equivalent to the S (with the multiplication of the respective learning weights) and d is the dimension of the feature extraction units. Multi-head attention first linearly projects the queries, keys and values to the given no. of the head (t) using different linear projections. Then these projections perform the scaled dot-product attention in parallel. Finally, these results of attention are concatenated and once again projected to get the new representation. Formally, the multi-head attention at head i can be computed by:

$$\begin{aligned} head_i &= Attention(SW_i^Q, SW_i^K, SW_i^V) \\ S^{'} &= W(head_1 \oplus head_2 \oplus, \ldots, \oplus head_t) \end{aligned} \quad (10)$$

where W_i^Q, W_i^K and W_i^V are the weight matrices.

3.7 Conditional Random Field Layer

In sequence labeling problem there is often a dependency between the successive labels. Instead of predicting the current label independently through softmax layer, we employ the CRF (Lafferty et al., 2001) layer, which takes care of the previous label to predict the current label. Firstly, the attentive feature at given time step t is projected to another space which has a dimension equal to the number of output tags. Mathematically, it can be formulated as follows:

$$o_t = W^S S_t^{'} + b^S \quad (11)$$

Thereafter, we calculate the score to predict a given label sequence y as follows:

$$score(y|X) = \sum_{t=1}^{n}(A_{t-1,t} + o_{t,y_t}) \quad (12)$$

where A is the transition score matrix. Finally, we select the tag sequence with highest score as follows:

$$\hat{y} = \arg\max_{y \in Y} score(y|x) \quad (13)$$

In decoding stage, we use Viterbi algorithm to compute the optimal tag sequence.

4 Experimental Details

4.1 Network Training

We have optimized two different losses to train our multi-task model. The first loss is task-specific loss of $\mathcal{L}_{task}$, which is specific for each task. Apart from task-specific loss, we also optimize the adversarial loss to train the network not correctly to predict the task.

For task-specific loss, we use negative log-likelihood objective as the loss function for each task. Given the total number of task T and N training samples (x_i, y_i) from task $t \in T$, the task loss $\mathcal{L}_{task}$ can be computed by the following equation:

$$\mathcal{L}_{task} = -\sum_{t=1}^{T}\sum_{i=1}^{N} \log p(\hat{y_i}^t | x_i^t) \quad (14)$$

The likelihood function $p(\hat{y_i}^t | x_i^t)$ can be computed by the following equation:

$$p(\hat{y_i}^t | x_i^t) = \frac{e^{score(\hat{y_i}^t | x_i^t)}}{\sum_{\overline{y} \in Y} e^{score(\overline{y_i}^t | x_i^t)}} \quad (15)$$

The $score(.)$ function is computed by the equation 12. The adversarial loss trains the shared feature extractor to generate the shared features such that the task discriminator layer cannot reliably recognize which task the input text comes from. The adversarial loss $\mathcal{L}_{adv}$ can be computed as follows:

$$\mathcal{L}_{adv} = \min_G \Big(\max_D \big(\sum_{t=1}^{T}\sum_{i=1}^{N} d_i^t log\big[D\big(G(x_i^t)\big)\big]\big) \Big) \quad (16)$$

where d_i^t is the gold label indicating the type of the current task and x_i^t is the i^{th} example of task t. The min-max optimization problem is addressed by the gradient reversal layer (Ganin and Lempitsky, 2015). The final loss of the model is defined by the following equation:

$$\mathcal{L} = \alpha \times \mathcal{L}_{task} + \beta \times \mathcal{L}_{adv} \quad (17)$$

where α and β are the scalar parameter.

4.2 Hyper-parameters

We use the pre-trained word embedding [2] from Pyysalo et al. (2013) of dimension 200. It is trained on the combination of PubMed and PMC biomedical texts with texts extracted from a recent English Wikipedia dump. We set the maximum length of input text as 44 and maximum character

[2] http://evexdb.org/pmresources/vec-space-models/

length of 10. The CNN based character embedding length of 100 is used in this experiment. The optimal hidden state dimension of GRU is set to be 100. We use 4 GRU layers to form the stacked GRU layer. The CNN layer uses the filter set: $\{2, 3, 4\}$. In multi-head attention layer, we use a total of 4 heads to compute the attentive representation. We set the dropout rate to 0.5. The batch size is set to 16 and value of loss weights α and β are set to be 0.8 and 0.2, respectively. The Adam Optimization (Kingma and Ba, 2015) method with a learning rate of 0.01 is used during training to optimize the network weights. The optimal values of hyper-parameters are achieved through the 10-fold cross validation experiment.

4.3 Datasets

We use three different ADR labeling datasets : PSB 2016 Social Media Shared Task for ADR Extraction dataset (Twitter), CADEC, and MEDLINE to evaluate our multi-task model performance. It is to be noted that our model is trained simultaneously on the different ADR datasets. The different datasets used in the experiment are as follows:

1. **Twitter dataset**: The first dataset, which we use is the Twitter dataset from PSB 2016 Social Media Shared Task for ADR Extraction task. It contains 572 tweets which are fully annotated for mentions of ADR, tweet ID, start and end offset, UMLS ID, annotated text span and the related drugs. We extracted the following three tags from this dataset: *ADR*, *Drugs*, and *Indications*.

2. **CADEC adverse drugs events dataset**: The another dataset, which we use is the CADEC adverse drugs event dataset. It contains a total of 1248 sentences containing different tags. Our model extract the following tags from CADEC Corpus: *ADR*, *Disease*, *Drug*, *Finding* and *Symptom*.

3. **MEDLINE ADR dataset**: This ADR corpus was released by Gurulingappa et al. (2012b). It was derived from the MEDLINE case reports[3]. This case report provides information about the symptoms, signs, diagnosis, treatment and follow-up of individual patients. This corpus contains 2972 documents with 20967 sentences. Out of which, 4272 sentences are annotated with names and relationships between drugs, adverse effects and dosages. Our model extract the *Drug* and *ADR* mentions in the sentences.

[3] https://www.nlm.nih.gov/bsd/indexing/training/PUB_050

5 Result and Analysis

We evaluate the pharmacovigilance labeling tasks in terms of Precision, Recall and F1-Score. Unlike the existing system, we evaluate the performance of our model, using the exact matching scheme, where a prediction sequence is counted as correct only if all the sequence labels are predicted correctly. We will begin by first describing the baselines models, followed by the results obtained from the proposed model and then present the analysis of the results.

5.1 Baselines

We compare our adversarial multi-task model with the following state-of-the-art baselines. It is to be noted that these baselines are re-implementation of the state-of-the-art methods for ADR extraction.

(1) ST-BLSTM: This is a single task model for ADR labeling with Bi-LSTM as sentence encoder. In our experiment, we build the individual model for each dataset.

(2) ST-CNN: This model is similar to baseline ST-BLSTM, but instead of using Bi-LSTM for sentence encoder, we use CNN with filters: $\{2, 3, 4\}$.

(3) CRNN: In this model CNN and LSTM are together used for sentence encoder (Huynh et al., 2016). We adopt the same architecture for ADR extraction by classifying each token of the sentence into a pre-defined set of tags.

(4) RCNN: This model is similar to the third baseline, but here we extract the LSTM feature first and then pass these features as the input to the CNN network.

(5) MT-BLSTM: It is a multi-task model (Chowdhury et al., 2018) with a shared Bi-LSTM layer across the task for sentence encoder and task-specific Bi-LSTM for each task. The final representation is obtained by concatenating shared and task-specific Bi-LSTM.

(6) MT-Atten-BLSTM: This baseline model (Chowdhury et al., 2018) is similar to the MT-BLSTM. The sentence encoder of this model is also equipped with the word level attention mechanism.

Models	Twitter			CADEC			MEDLINE		
	P	R	F1	P	R	F1	P	R	F1
ST-BLSTM	57.7	56.8	57.3	52.9	49.4	51.1	71.65	72.19	71.91
ST-CNN	63.8	65.8	67.1	39.7	42.7	42.0	66.88	73.81	70.17
CRNN (Huynh et al., 2016)	61.1	62.4	64.9	49.5	46.9	48.2	71.0	77.3	75.5
RCNN (Huynh et al., 2016)	57.6	58.7	63.6	42.4	44.9	43.6	73.5	72.0	74.0
MT-BLSTM (Chowdhury et al., 2018)	65.57	61.02	63.19	60.50	55.16	57.62	72.72	75.49	74.0
MT-Atten-BLSTM (Chowdhury et al., 2018)	62.26	69.62	65.73	56.63	60.0	58.27	75.08	81.06	77.95
Proposed Model	68.78	70.81	69.69	64.33	67.03	65.58	81.97	82.61	82.18

Table 1: Result comparison of the proposed method with the state-of-art baseline methods. Here, 'P', 'R', 'F1' represents Precision, Recall and F1-Score. The results on CADEC and MEDLINE are on 10-fold cross validation; for the twitter dataset, we use the train and test sets as provided by the PSB 2016 shared task.

Model Components	Twitter	CADEC	MEDLINE
Proposed Model	69.69	65.58	82.18
– Character Embedding	67.63 (2.06 ↓)	56.10 (9.48 ↓)	76.34 (5.84 ↓)
– Multi-head Attention	68.65 (1.04 ↓)	60.51 (5.07 ↓)	77.71 (4.47 ↓)
– Adversarial Learning	68.11 (1.58 ↓)	58.57 (7.01 ↓)	71.21 (10.97 ↓)

Table 2: Ablation study on all the dataset. The values in the bracket shows the absolute decrements (↓) in the proposed model by removing the respective component. It shows the contribution (in terms of model performance in F1 Score) of that component in our proposed model.

5.2 Results

The extensive results of our proposed model with comparisons to the state-of-the-art baselines techniques are reported in Table 1. Our proposed model outperforms the state-of-the-art baselines techniques by fair margins in terms of precision, recall and F1-Score for all the datasets. In our first experiment, we train two models (i.e. Single-Task BLSTM and Multi-Task BLSTM) to analyze the effect of the multi-task model (MT-BLSTM) over a single task model (ST-BLSTM). On all the three datasets, we can visualize from Table 1 that, the multi-task framework with its sharing scheme can help in boost the performance of the system. We observe the performance improvement of 5.89, 6.52 and 2.09 F1-Score points on Twitter, CADEC, and MEDLINE dataset, respectively. The similar improvement is also observed in terms of precision and recall.

In comparison to the baseline 5 model, our proposed method achieve the improvement of 6.5, 7.96, and 8.18 F1-Score points on Twitter, CADEC, and MEDLINE dataset, respectively. This shows the robustness of our proposed multi-task method. We also compare our proposed system with MT-Atten-BLSTM model. The results show the performance improvement of 3.96, 7.31, and 4.23 F1 Score points for Twitter, CADEC and MEDLINE dataset, respectively. The improvements over all the baselines methods are statistically significant as $p < 0.05$.

5.3 Ablation Study

To analyze the impact of various component of our model, we perform the ablation study (c.f. Table-2) by removing one component from the proposed model and evaluate the performance on all the three datasets. Character embedding is found to be the most crucial component on Twitter, and CADEC datasets as both of these datasets are from the social media text and carry the nature of the short text and out of vocabulary words.

To prove our hypothesis (introduction of adversarial learning in the multi-task framework can make shared space independent of the task invariant features), we exclude the adversarial loss from our proposed framework. We could see a significant decline in performance. This depicts that making the task shared space free from the contamination of task-specific feature, can significantly improve the performance of the system. Removal of the multi-head attention also lead to drop of an average 4% F1-Score points across all the datasets.

6 Analysis

To get a deeper insight into how our multi-task model performs over the state-of-the-art multi-task baseline model, we sample few sentences from all the three datasets. In the Table-3, we demonstrate the capability of our model in correctly predicting all the labels, while the MT-LSTM and MT-LSTM-atten make the incorrect prediction. In the sentence 1 due to the sharing scheme, bipolar was correctly labeled as Indication.

Sentence 1	fluoxetine	and	quet	combo	zombified	me	ahh	the	med	merrygoround	bipolar
Actual Labels	B-Drug	O	B-Drug	O	B-ADR	O	O	O	O	O	B-Indication
MT-LSTM	B-Drug	O	B-Drug	O	O	O	O	O	O	O	O
MT-LSTM-Atten	B-Drug	O	B-Drug	O	B-ADR	O	O	O	O	O	O
Proposed Approach	B-Drug	O	B-Drug	O	B-ADR	O	O	O	O	O	B-Indication
Sentence 2	clozapine-induced	tonic-clonic	seizure	managed	with	valproate	implication	for	clinical	care	
Actual Labels	B-Drug	B-ADR	I-ADR	O	O	O	O	O	O	O	
MT-LSTM	B-Drug	O	O	O	O	O	O	O	O	O	
MT-LSTM-Atten	B-Drug	O	B-ADR	O	O	B-ADR	O	O	O	O	
Proposed Approach	B-Drug	B-ADR	I-ADR	O	O	O	O	O	O	O	

Table 3: Comparison of the predictions of the proposed approach with the baseline models.

	Sentence 1	too	much	zoloft	and	seroquel	to	get	the	horn	my	life	is	lie			
Type-1	**Actual**	O	O	B-Drug	O	B-Drug	O	O	O	B-ADR	I-ADR	I-ADR	O	O			
	Predicted	O	O	B-Drug	O	B-Drug	O	O	O	O	O	O	O	O			
	Sentence 2	pain	in	upper	right	arm	could	not	sleep	on	it	or	move	it	behind	my	back
Type-2	**Actual**	B-ADR	I-ADR	I-ADR	I-ADR	I-ADR	O	O	O	O	O	O	O	O	O	O	O
	Predicted	B-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR
	Sentence 3	terrible	joint	pain	could	not	move	shoulder	hip	hurt							
Type-3	**Actual**	O	B-ADR	B-ADR	B-ADR	I-ADR	I-ADR	I-ADR	B-ADR	I-ADR							
	Predicted	B-ADR	I-ADR	I-ADR	B-ADR	I-ADR	I-ADR	I-ADR	I-ADR	I-ADR							

Table 4: Exemplar description of various types of error. Here, Type-1 represent the error due '*Presence of implicit mention*'. Type-2 represent the error due to '*Issue in annotation*' and Type-3 represents the error of type '*Boundary detection problem*'.

In the sentence 2, we observe that, only MT-LSTM-Atten model is able to predict the partial ADR (i.e. seizure instead of tonic-clonic seizure.), while our model is able to predict the full ADR phrase correctly.

6.1 Error Analysis

In this subsection, we analyze the different sources of errors which lead to mis-classification. We closely study the false positive and false negative instances and come up with the following observations:

(1) Presence of implicit mention: We observe that in the Twitter dataset user often tends to use very implicit and creative language to describe their adverse drug reaction. For e.g., in the sentence-1 of Table-4, user describes his ADR as '*horn my life*' by taking drug (*zoloft and seroquel*).

(2) Issue in annotation: For the CADEC dataset, we observe some of the sentences are not completely tagged. For e.g., in the sentence-2 of Table-4, here '*could not sleep*', '*move it behind my back*' is also an ADR, in addition to '*pain in upper right arm*'. However, the first two ADRs are not labeled in the dataset.

(3) Boundary detection problem: We also observe that, our system sometimes fails to detect the proper boundary. This might be because of the task sharing feature, which learns the feature distributions across the dataset which may not be correct for the given dataset as shown in sentence-3 of Table-4.

7 Conclusion

In this paper, we have proposed an end-to-end multi-task framework that provides a unified solution for pharmacovigilance mining. We have utilized an adversarial training based multi-task framework, which ensures that task-specific and task shared features are not contaminated. We evaluated this framework on three benchmark pharmacovigilance datasets. Our results demonstrate the capability of our model across all the datasets. In future, we would like to assist the model with multiple linguistic aspects of social media text like figurative languages.

Acknowledgement

Sriparna Saha and Asif Ekbal gratefully acknowledge the Young Faculty Research Fellowship (YFRF) Award, supported by Visvesvaraya PhD scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) for carrying out this research. Authors acknowledge "Shusrut: ezDI Research Lab on Health Informatics", Department of Computer Science and Engineering, IIT Patna, India.

References

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, Kayo Waki, and Kazuhiko Ohe. 2010. Extraction of adverse

drug effects from clinical records. *Studies in health technology and informatics*, 160 Pt 1:739–43.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Rachel E Behrman, Joshua S Benner, Jeffrey S Brown, Mark McClellan, Janet Woodcock, and Richard Platt. 2011. Developing the sentinel systema national resource for evidence development. *New England Journal of Medicine*, 364(6):498–499.

Adrian Benton, Lyle Ungar, Shawndra Hill, Sean Hennessy, Jun Mao, Annie Chung, Charles E Leonard, and John H Holmes. 2011. Identifying potential adverse effects using the web: A new approach to medical hypothesis generation. *Journal of biomedical informatics*, 44(6):989–996.

Shaika Chowdhury, Chenwei Zhang, and Philip S. Yu. 2018. Multi-task pharmacovigilance mining from social media posts. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 117–126, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Clark C Freifeld, John S Brownstein, Christopher M Menone, Wenjie Bao, Ross Filice, Taha Kass-Hout, and Nabarun Dasgupta. 2014. Digital drug safety surveillance: monitoring pharmaceutical products in twitter. *Drug safety*, 37(5):343–350.

Carol Friedman. 2009. Discovering novel adverse drug events using natural language processing and mining of the electronic health record. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 1–5. Springer.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1180–1189.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Harsha Gurulingappa, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2011. Identification of adverse drug event assertive sentences in medical case reports. In *First international workshop on knowledge discovery and health care management (KD-HCM), European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD)*, pages 16–27.

Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. 2012a. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3(1):15.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012b. 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):885–892.

Katja M Hakkarainen, Khadidja Hedna, Max Petzold, and Staffan Hägg. 2012. Percentage of patients with preventable adverse drug reactions and preventability of adverse drug reactions–a meta-analysis. *PloS one*, 7(3):e33236.

Rave Harpaz, William DuMouchel, Nigam H Shah, David Madigan, Patrick Ryan, and Carol Friedman. 2012a. Novel data-mining methodologies for adverse drug event discovery and analysis. *Clinical Pharmacology & Therapeutics*, 91(6):1010–1021.

Rave Harpaz, Santiago Vilar, William DuMouchel, Hojjat Salmasian, Krystl Haerian, Nigam H Shah, Herbert S Chase, and Carol Friedman. 2012b. Combing signals from spontaneous reports and electronic health records for detection of adverse drug reactions. *Journal of the American Medical Informatics Association*, 20(3):413–419.

Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. 2016. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887, Osaka, Japan. The COLING 2016 Organizing Committee.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2741–2749. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Robert Leaman, Laura Wojtulewicz, Ryan Sullivan, Annie Skariah, Jian Yang, and Graciela Gonzalez. 2010. Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the 2010 workshop on biomedical natural language processing*, pages 117–125. Association for Computational Linguistics.

Kathy Lee, Ashequl Qadir, Sadid A. Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. 2017. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 705–714, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Hui Li, Xiao-Jing Guo, Xiao-Fei Ye, Hong Jiang, Wen-Min Du, Jin-Fang Xu, Xin-Ji Zhang, and Jia He. 2014. Adverse drug reactions of spontaneous reports in shanghai pediatric population. *PLoS One*, 9(2):e89829.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada. Association for Computational Linguistics.

Azadeh Nikfarjam and Graciela H Gonzalez. 2011. Pattern mining for extraction of mentions of adverse drug reactions from user comments. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1019. American Medical Informatics Association.

Azadeh Nikfarjam, Abeed Sarker, Karen Oconnor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681.

Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek V. Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual LSTM networks. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2923–2934.

S Pyysalo, F Ginter, H Moen, T Salakoski, and S Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*, pages 39–44.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

Abeed Sarker, Azadeh Nikfarjam, and Graciela Gonzalez. 2016. Social media mining shared task workshop. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 581–592. World Scientific.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc.

Gabriel Stanovsky, Daniel Gruhl, and Pablo Mendes. 2017. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 142–151.

Janet Sultana, Paola Cutroneo, and Gianluca Trifirò. 2013. Clinical and economic burden of adverse drug reactions. *Journal of pharmacology & pharmacotherapeutics*, 4(Suppl1):S73.

Elena Tutubalina and Sergey Nikolenko. 2017. Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews. *Journal of Healthcare Engineering*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Xiaoyan Wang, George Hripcsak, Marianthi Markatou, and Carol Friedman. 2009. Active computerized pharmacovigilance using natural language processing, statistics, and electronic health records: a feasibility study. *Journal of the American Medical Informatics Association*, 16(3):328–337.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhifei Zhang, JY Nie, and Xuyao Zhang. 2016. An ensemble method for binary classification of adverse drug reactions from social media. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*.