Co-training for Extraction of Adverse Drug Reaction Mentions from Tweets

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adverse drug reactions (ADRs) are one of the leading causes of mortality in health care. Current ADR surveillance systems are often associated with a substantial time lag before such events are officially published. On the other hand, online social media such as Twitter contain information about ADR events in real-time, much before any official reporting. Current state-of-the-art methods in ADR mention extraction use Recurrent Neural Networks (RNN), which typically need large labeled corpora. Towards this end, we propose a semi-supervised method based on co-training which can exploit a large pool of unlabeled tweets to augment the limited supervised training data, and as a result enhance the performance. Experiments with $\sim$0.1M tweets show that the proposed approach outperforms the state-of-the-art methods for the ADR mention extraction task by $\sim$5% in terms of F1 score.

1 Introduction

Estimates show that Adverse Drug Reactions (ADRs) are the fourth leading cause of deaths in the United States ahead of cardiac diseases, diabetes, AIDS and other fatal diseases[1]. Hence, it necessitates the monitoring and detection of such adverse events to minimize the potential health risks. Typically, post-marketing drug safety surveillance (also called as pharmacovigilance) is conducted to identify ADRs after a drug's release. Such surveys rely on formal reporting systems such as Federal Drug Administration's Adverse Event Reporting System (FAERS)[2]. However, often a large fraction ($\sim$94%) of the actual ADR instances are under-reported in such systems [9]. Social media presents a plausible alternative to such systems, given its wide userbase. Twitter, a popular social media platform has 300M+ active users[3]. A recent study [6] shows that Twitter has three times more ADRs reported as compared to FAERS.

Earlier work in this direction focused on feature based pipeline followed by a sequence classifier [12]. More recent works are based on Deep Neural Networks [4, 13]. Deep learning based methods [5, 11] typically rely on the presence of a large annotated corpora, due to their large number of free parameters. Due to the high cost associated with tagging ADR mentions in a social media post and limited availability of labeled datasets, it is hard to train a deep neural network effectively for such a task. In this work, we attempt to address this problem and propose a novel semi-supervised method based on co-training [2] which can harness a large pool of unlabeled related tweets, which are more economical to collect than ADR annotated tweets.

[1] https://ethics.harvard.edu/blog/new-prescription-drugs-major-health-risk-few-offsetting-advantages
[2] http://bit.ly/2xnu7pE
[3] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

2 Approach

In this section, we first define the ADR extraction problem concretely. Then we briefly discuss the supervised ADR extraction method, and then propose our semi-supervised co-training method.

2.1 Problem Definition

The problem of ADR extraction can be defined as follows. Given a social media post in the form of word sequence $x = x_1....,x_n$, where n is the maximum sequence length, predict an output sequence $y_1, ..., y_n$, where each y_i is encoded using standard sequence labeling encoding scheme such as the IO encoding similar to that used in [4].

2.2 Co-training Method for ADR Extraction

Supervised ADR Extraction: We choose the model described in [4] for modeling the ADR extraction task. Given an input word sequence x, a bi-directional LSTM transducer (bi-LSTM) [8] is employed to capture complex sequential dependencies. Formally, at each time-step t, the LSTM transducer attempts to model the task as follows.

$$h_t = \text{bi-LSTM}(e_t, h_{t-1}) \tag{1}$$

where $h_t \in \mathcal{R}^{(2\times d_h)}$, is the hidden unit representation of the bi-LSTM with d_h being the hidden unit size. Since it is a concatenation of hidden units of a forward sequence LSTM and a backward sequence LSTM, its overall dimension is $2 \times d_h$. e_t is the embedding vector corresponding to the input word x_t extracted from a pre-trained word embedding lookup table.

$$y_t = \text{softmax}(W \times h_t + b) \tag{2}$$

where $y_t \in \mathcal{R}^{d_l}$, is the output vector at each time-step which encodes the probability distribution over the number of possible output labels (d_l) at each time-step of the sequence. $W \in \mathcal{R}^{d_l * d_h}$ and $b \in \mathcal{R}^{d_l}$ are weight vectors for the affine transformation. Finally, the loss function for the task is defined as follows.

$$L_{\text{ADR}} = -\sum_{t=1}^{n}\sum_{i=1}^{d_l} \hat{y_{t_i}} \log y_{t_i} \tag{3}$$

where $\hat{y_t}$ is the one-hot representation of the actual label at time-step t.

Semi-Supervised Co-training: Algo. 1 outlines the method for semi-supervised co-training. Co-training [2] requires two feature views of the dataset, which in our case are: (1) word2vec embeddings trained on a generic tweet corpus [7] followed by a bi-LSTM feature extractor and (2) word2vec embeddings trained on domain specific (drug-related) tweet corpus (described in the Section 3) followed by a bidirectional Gated Recurrent Unit (bi-GRU) transducer [3].

At each step of co-training, the transducers M_1 and M_2 are trained on their respective views minimizing the ADR training loss (Line 4 to 5 of Algo. 1). Each sample from the unlabeled example pool is scored using a scoring function computed as follows. First, the current transducer is used to decode/infer output label distribution for each word in the unlabeled sample. For each word in the output sequence, we simply choose the output label which has the maximum probability. We filter out the data sample if the transducer does not output even a single ADR label for any word in the sample. If there is at least one word labeled as ADR, we compute the score for the sample as the multiplication of the ADR probabilities for the ADR-labeled words in the sample normalized by the number of ADR words. If this confidence score of the sample is greater than some pre-defined threshold τ, the sample is added to the training set of the other transducer along with its output labels as generated by the transducer (Lines 7, 8, 10 and 11). Due to this cross-exchange of training data, both transducers work in synergy and learn from mistakes of each other.

3 Experiments

In this section, we discuss details of the datasets, implementation details and experimentation results.

Algorithm 1 Co-training Method for ADR Extraction

Input U: Large collection of unlabeled tweets
τ : Threshold for co-training
D^1_{ADR}, D^2_{ADR} : Two views of the ADR annotated data
Output Model parameters θ^{LSTM}, θ^{GRU}

$T^1, T^2 \leftarrow D^1_{ADR}, D^2_{ADR}$
Initialize model parameters, θ^{LSTM}, θ^{GRU} randomly.
while *(stopping criteria is not met)* **do**
    $M^1 \leftarrow$ train bi-LSTM on T^1 (minimize L^1_{ADR})
    $M^2 \leftarrow$ train bi-GRU on T^2 (minimize L^2_{ADR})
    for $i \leftarrow 1, |U|$ **do**
        if $M^1.score(U_i) \geq \tau$ **then**
            $T^2 \leftarrow T^2 \cup \{U_i\}$
            $U \leftarrow U - U_i$
        if $M^2.score(U_i) \geq \tau$ **then**
            $T^1 \leftarrow T^1 \cup \{U_i\}$
            $U \leftarrow U - U_i$

3.1 Datasets

The statistics of the datasets are presented in Table 1. We use the following datasets:

1) Twitter ADR We use the Twitter dataset, *Twitter ADR* described in [4]. It contains 957 tweets posted between 2007 and 2010, with mention annotations of ADR and some other medical entities. Due to Twitter's license agreement, authors released only tweet ids with their corresponding mention span annotations. At the time of collection of the original tweets using Twitter API, we were able to collect only 639 tweets.

2) TwiMed We use the second Twitter dataset, *TwiMed* described in [1]. It contains 1000 tweets with mention annotation of Symptoms from drug (ADR) and other mention annotations posted in 2015. Due to Twitter's license agreement, we were able to extract 663 tweets only.

3) Unlabeled Tweets For the unlabeled tweets used for semi-supervised learning, we collected tweets using the keywords as drug-names and ADR lexicon publicly available[4]. This filtering step ensures that all collected tweets have at least one drug-name occurrence and one ADR phrase. The tweets were posted in 2015.

Dataset	No. tweets	No. ADR
Twitter ADR	639	27,333
TwiMed	663	28,262
Unlabeled Tweets	1,00,000	-

Table 1: Dataset Statistics

3.2 Implementation Details

For implementation of the model, we use the popular Python deep learning toolkits: Keras[5] and TensorFlow[6]. Training data for each fold is divided according to 90:10% train-validation split.

Pre-processing As part of text pre-processing, all HTML links and user mentions are normalized to a single token respectively. Special characters and emoticons are removed, and each tweet is padded with the maximum tweet length in the corpus.

Hyper-parameter settings for the two views: The hyper-parameter settings for the two views as required by the co-training method are as follows.

[4]http://diego.asu.edu/downloads
[5]https://keras.io/
[6]https://www.tensorflow.org/

Method	Precision	Recall	F1-score
Baseline [4]	0.7067 ± 0.057	0.7207 ± 0.074	0.7102 ± 0.049
Baseline with adam	0.7065 ± 0.058	0.7576 ± 0.083	0.7272 ± 0.051
KB-Embedding Baseline [13]	0.7171 ± 0.058	0.7713 ± 0.091	0.7397 ± 0.055
Co-training (5k)	0.7247 ± 0.056	0.7770 ± 0.082	0.7488 ± 0.063
Co-training (10k)	0.7288 ± 0.041	$\mathbf{0.8238 \pm 0.064}$	0.7719 ± 0.040
Co-training (25k)	0.7181 ± 0.035	0.8005 ± 0.048	0.7561 ± 0.031
Co-training (50k)	0.7207 ± 0.034	0.7870 ± 0.042	0.7516 ± 0.029
Co-training (75k)	0.7478 ± 0.062	0.8033 ± 0.053	0.7730 ± 0.047
Co-training (100k)	$\mathbf{0.7514 \pm 0.053}$	0.8045 ± 0.056	$\mathbf{0.7754 \pm 0.042}$

Table 2: Experimental Results for Twitter ADR dataset (along with Std. Deviation)

View 1: For the first view we use bi-LSTM transducer, with the hyper-parameter setting similar to the one reported in [4]. Word embedding dimension is set to 400. The hidden unit dimension (d_h) is set to 500.

View 2: For the second view, we use bi-GRU transducer with input as word2vec word embeddings trained on the unlabeled drug-related tweets described in the previous section. The word embedding dimension is set to 300. The number of output units (d_l) is 4. We use adam [10] as optimizer with a learning rate of 0.001 and a batch size of 32.

Co-training Parameters: For the co-training methods, confidence threshold value is empirically set to 0.5. The stopping criteria for the co-training kicks in when the number of iterations reaches 5 or if the unlabeled tweets pool is exhausted, whichever occurs first. The number of epochs are set to a maximum with 25, with early-stopping employed if validation loss drops for more than 3 epochs.

3.3 Results

The results of various methods are presented in Tables 2 and 3 for the Twitter ADR and the TwiMed datasets respectively. For the ADR task, to encode the output labels we use the IO encoding scheme where each word is labeled with one of the following labels: (1) I-ADR (ADR mention), (2) I-Other (mention category other than ADR), (3) O (others), (4) PAD (padding token). Since our entity of interest is ADR, we report the results on ADR only. An example tweet annotated with IO-encoding is as follows. *@BLENDOS$_O$ Lamictal$_O$ and$_O$ trileptal$_O$ and$_O$ seroquel$_O$ of$_O$ course$_O$ the$_O$ seroquel$_O$ I$_O$ take$_O$ in$_O$ severe$_O$ situations$_O$ because$_O$ weight$_{I\text{-}ADR}$ gain$_{I\text{-}ADR}$ is$_O$ not$_O$ cool$_O$*. For performance evaluation, we use approximate-matching [14], which is used popularly in biomedical entity extraction tasks [4, 12]. We report the F1-score, Precision and Recall computed using approximate matching as follows.

$$\text{Precision} = \frac{\text{\#ADR approx. matched}}{\text{\#ADR spans predicted}} \tag{4}$$

$$\text{Recall} = \frac{\text{\#ADR approx. matched}}{\text{\#ADR spans in total}} \tag{5}$$

The F1-score is the harmonic-mean of the Precision and Recall values. All results are reported using 10-fold cross-validation along with the standard deviation across the folds. Our baseline methods are bi-LSTM transducer [4] with traditional word embeddings and the current state-of-the-art bi-LSTM transducer which used traditional word embeddings augmented with knowledge-graph based embeddings [13]. For both the datasets, it should be noted that Cocos et al. [4] used RMSProp as an optimizer, and since we are using adam for all our methods, so for a fair comparison we also report the baseline results with adam. The corresponding results are reported in the first two rows of Tables 2 and 3 respectively. It is clear that re-implementation with adam optimizer enhances the performance, which is consistent with the general consensus around adam optimizer. The KB-embedding baseline [13] replaces word embeddings of the medical entities in the sentence with the corresponding embeddings learned from a knowledge-base. The corresponding results can be seen in row 3. It is clear that adding KB-based embeddings enhances the performance over the baseline, due to the external knowledge added in the form of KB embeddings.

Method	Precision	Recall	F1-score
Baseline [4]	0.6120 ± 0.116	0.5149 ± 0.099	0.5601 ± 0.100
Baseline with adam	0.6281 ± 0.094	0.5614 ± 0.110	0.5859 ± 0.079
KB-Embedding Baseline [13]	0.5960 ± 0.081	0.6144 ± 0.068	0.6042 ± 0.060
Co-training (5k)	$\mathbf{0.5806} \pm 0.093$	0.6746 ± 0.078	$\mathbf{0.6192 \pm 0.066}$
Co-training (10k)	0.5484 ± 0.092	0.6355 ± 0.113	0.5851 ± 0.090
Co-training (25k)	0.5774 ± 0.082	0.6425 ± 0.076	0.6051 ± 0.066
Co-training (50k)	0.5420 ± 0.054	0.6342 ± 0.061	0.5836 ± 0.053
Co-training (75k)	0.5525 ± 0.059	$\mathbf{0.6875 \pm 0.069}$	0.6110 ± 0.056
Co-training (100k)	0.5548 ± 0.064	0.6786 ± 0.058	0.6081 ± 0.048

Table 3: Experimental Results for TwiMed dataset (along with Std. Deviation)

The results for our methods are presented from row 4 onwards. It is clear that the co-training method outperforms the baseline by a significant margin. It clearly indicates the efficacy of semi-supervised learning when the labeled data is scarce.

Effect of Unlabeled Data Size: We also analyze the effect of the size of the unlabeled tweet dataset on the method's performance. The results are presented from row 4 onwards. The results are fairly constant as unlabeled data size is varied, indicating the robustness of the method.

4 Conclusion and Future Work

In this paper, we proposed a semi-supervised co-training based learning based methods to tackle the problem of labeled data scarcity for adverse drug reaction mention extraction task. Our method uses large unlabeled drug related tweets to augment the limited existing ADR extraction datasets providing superior results in comparison to pure supervised learning based methods. We analyzed the method on two popular ADR extraction datasets, and it demonstrates superior results as compared to the state-of-the-art methods in ADR extraction.

References

[1] N. Alvaro, Y. Miyao, and N. Collier. TwiMed: Twitter and PubMed Comparable Corpus of Drugs, Diseases, Symptoms, and Their Relations. *JMIR Public Health and Surveillance*, 3(2), 2017.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[3] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[4] A. Cocos, A. G. Fiks, and A. J. Masino. Deep Learning for Pharmacovigilance: Recurrent Neural Network Architectures for Labeling Adverse Drug Reactions in Twitter Posts. *JAMIA*, page ocw180, 2017.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (almost) from Scratch. *JMLR*, 12(Aug):2493–2537, 2011.

[6] C. C. Freifeld, J. S. Brownstein, C. M. Menone, W. Bao, R. Filice, T. Kass-Hout, and N. Dasgupta. Digital Drug Safety Surveillance: Monitoring Pharmaceutical Products in Twitter. *Drug Safety*, 37(5):343–350, 2014.

[7] F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle. Multimedia Lab@ ACL W-Nut NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. *ACL-ICJNLP*, 2015:146–153, 2015.

[8] A. Graves. Sequence Transduction with Recurrent Neural Networks. *CoRR*, abs/1211.3711, 2012.

[9] L. Hazell and S. A. Shakir. Under-reporting of Adverse Drug Reactions: A Systematic Review. *Pharmacoepidemiology and Drug Safety*, 14:S184–S185, 2005.

[10] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

[11] Y. LeCun, Y. Bengio, and G. Hinton. Deep Learning. *Nature*, 521(7553):436–444, 2015.

[12] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, and G. Gonzalez. Pharmacovigilance from Social Media: Mining Adverse Drug Reaction Mentions using Sequence Labeling with Word Embedding Cluster Features. *JAMIA*, 22(3):671–681, 2015.

[13] G. Stanovsky, D. Gruhl, and P. N. Mendes. Recognizing Mentions of Adverse Drug Reaction in Social Media Using Knowledge-Infused Recurrent Models. In *EACL*, pages 142–151, 2017.

[14] R. T.-H. Tsai, S.-H. Wu, W.-C. Chou, Y.-C. Lin, D. He, J. Hsiang, T.-Y. Sung, and W.-L. Hsu. Various Criteria in the Evaluation of Biomedical Named Entity Recognition. *BMC Bioinformatics*, 7(1):92, 2006.