Improving RNN with Attention and Embedding for Adverse Drug Reactions

Chandra Pandey*
Kings College London, IoPPN
Denmark Hill
London SE5 8AF
chandra.pandey@kcl.ac.uk

Zina Ibrahim
Kings College London, IoPPN
Denmark Hill
London SE5 8AF
zina.ibrahim@kcl.ac.uk

Honghan Wu
Kings College London, IoPPN
Denmark Hill
London SE5 8AF
honghan.wu@kcl.ac.uk

Ehtesham Iqbal
Kings College London, IoPPN
Denmark Hill
London SE5 8AF
ehtesham.iqbal@kcl.ac.uk

Richard Dobson
Kings College London, IoPPN
Denmark Hill
London SE5 8AF
richard.j.dobson@kcl.ac.uk

ABSTRACT

Electronic Health Records (EHR) narratives are a rich source of information, embedding high-resolution information of value to secondary research use. However, because the EHRs are mostly in natural language free-text and highly ambiguity-ridden, many natural language processing algorithms have been devised around them to extract meaningful structured information about clinical entities. The performance of the algorithms however, largely varies depending on the training dataset as well as the effectiveness of the use of background knowledge to steer the learning process.

In this paper we study the impact of initializing the training of a neural network natural language processing algorithm with pre-defined clinical word embeddings to improve feature extraction and relationship classification between entities. We add our embedding framework to a bi-directional long short-term memory (Bi-LSTM) neural network, and further study the effect of using attention weights in neural networks for sequence labelling tasks to extract knowledge of Adverse Drug Reactions (ADRs). We incorporate unsupervised word embeddings using Word2Vec and GloVe from widely available medical resources such as Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) II corpora, Unified Medical Language System (UMLS) as well as embed pharmaco lexicon from available EHRs. Our algorithm, implemented using two datasets, shows that our architecture outperforms baseline Bi-LSTM or Bi-LSTM networks using linear chain and Skip-Chain conditional random fields (CRF).

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; **Information extraction**; **Neural networks**;

KEYWORDS

Recurrent Neural Networks, Named Entity Recognition, Adverse Drug Reactions

ACM Reference format:
Chandra Pandey, Zina Ibrahim, Honghan Wu, Ehtesham Iqbal, and Richard Dobson. 2017. Improving RNN with Attention and Embedding for Adverse Drug Reactions. In *Proceedings of DH '17, London, United Kingdom, July 02-05, 2017,* 5 pages.
https://doi.org/http://dx.doi.org/10.1145/3079452.3079501

1 INTRODUCTION

During recent years, the importance of the information embedded within electronic health records (EHRs) for understanding disease and treatment has been recognized [14]. As a result, many studies focus on the identification of named entities and relationships from clinical notes. Apart from rule-based approaches, whereby the reliance is on hand-crafted rules and domain-dependent dictionaries to obtain performance [9], many machine learnings have been applied such as support vector machines (SVMs) and conditional random fields (CRFs) and [12] among a few.

Regardless of the model chosen, it must deal with the challenges imposed by the type of text embedded within clinical notes. In contrast to general biomedical text, which is intended to communicate research results, electronic health records (EHRs) pose a unique set of challenges for named entity recognition and relationship extraction. First, as opposed to the unambiguous and explicitly codified content contained within the general biomedical text, EHR narratives are highly heterogeneous in their content (ranging from discharge summaries to progress notes to consultations). Moreover, the general scarcity of EHRs for natural language processing tools development and testing has created a lag in the progress of biomedical entity recognition from free-text clinical narratives as opposed to general biomedical text from the scientific literature. Finally, there is a significant discrepancy between the performance

*All authors have contributed equally to the research.

DH '17, July 02-05, 2017, London, United Kingdom

ACM ISBN 978-1-4503-5249-9/17/07…$15.00
https://doi.org/http://dx.doi.org/10.1145/3079452.3079501

of classifying simple (usually one-word or standardized format) entities and relationships such as drug name, dosage, frequency, and severity, as opposed to more sophisticated concepts such as ADEs. With ADEs, the concepts to be extracted and their connecting relationships a) frequently span several, and possibly non-consecutive locations in a given sentence and b) their is highly dependent on the context in which they occur [10]. As a results, the performance of machine learning algorithms has traditionally ranged between F-scores of 0.69 to 0.88.

In this paper, we evaluate the use of word embeddings on the performance of machine learning algorithms in annotating EHR text. Our focus is on constructing efficient word embeddings form known clinical entities (drug names, diseases, ADR positive-negative terms, uncommon abbreviations and rare terms) derived from EHR text. We design a neural network model which uses the embeddings to enhance context evaluation of ADE terms. Mainly, the embeddings are provided as input to a Bi-Directional Long Short-term Memory (LSTM) layer of a Recurrent Neural Network (RNN) to model the context information of each word.

We also model long term and short phrase dependencies in the text to enhance context identification. Spanning over a context window to form the closest context vectors will result in better indications of relationships between labels. We use attention mechanism inspired by works from [2] on top of the RNN to output *attention weights* and generate them at every step. On top of Bi-LSTM with attention mechanism, we use a CRF layer to jointly decode labels for the whole sentence to detect ADE presence.

2 BACKGROUND

2.1 Named Entity Recognition

NER is the ability to recognize references to entities and classify them into semantic categories. In biomedical domain most of the NER systems such as MedLEE [4], MetaMap [1] and cTAKES [19] are rule based and rely on medical dictionaries. A move in the direction of machine learning has led to CRF models that that can learn to recognize the entities automatically, with F-Scores ranging from 0.69 to 0.88.

Named entity recognition in biomedical domain have seen the use of annotated biomedical corpus. CRF tagging models have been used to extract classify entity domain such as names of protein, genes, DNA and the like. LSTM models have also been used for NER on BioCreative corpus. [5] extracted ADE from Medline corpus using many biomedical dictionaries.

2.2 Word Embeddings

Word embeddings are vector representations of words that have played an important role in biomedical named entity recognition as an application of deep learning techniques. Word2Vec [15], the prime example of word embeddings, has been investigated for clinical sentiment analysis, relation extraction and named entity recognition. In event extraction, [13] applied word embeddings for BioNLP event extraction tasks, [16] conducted biological event trigger identification with embedding enabled neural network model and [6] demonstrated unsupervised methods that exploit co-occurrence of information to model in a vector space to improve predictive performance. In our approach we train the Word2Vec word embeddings from PubMed articles, MIMIC datasets, Wikipedia articles, Drug-Disease pairs from EHR text, abbreviations and positive-negative phrases compiled from our medical corpus.

2.3 Bi-LSTM (Baseline)

LSTM networks are a type of RNN which have been beneficial in sequence labeling tasks as they can make use of the past hidden states ht for a stipulated period of time. However, it is unable to know the future states which would be useful in the labeling task. Bi-LSTM networks on the other hand store the hidden states from a forward and backward pass. We can make use of the past and future states to label the sequence at hand. In the baseline model we use the word embedding as input, a Bi-LSTM neural network and a Softmax Output layer. The input text is tokenized in a sequence of tokens.

2.4 CRF

CRF models have been widely used in sequence labeling tasks when the labels of surrounding neighbours have to be jointly decoded for a given sentence. [17] have used CRF classifier for concept extraction in social media text. CRFSuite, an implementation of [18] provides a fast and simple interface for training and modifying input features. The CRF classifier is trained on annotated mentions of ADR and it classifies tokens in sentences. If we represent $z = z_1....z_n$, as the embedding vector for the ith word in the input sequence and $y = y_1, ..., y_n$ as the sequence of labels for z from a set of sequences $Y(z)$. For a linear chain CRF, the family of conditional probility $p(\boldsymbol{y} \mid \boldsymbol{z}; \boldsymbol{W}, \boldsymbol{b})$ can be written as follows:

$$p(y \mid z) = \frac{1}{Z(x)} \prod_{i=1}^{n} \Psi_i(y_i, y_{i-1}, z) \tag{1}$$

Here Z is the partition function used for normalizing the local factor function Ψ_t. Thus Z can be written as:

$$Z = \sum_{y' \in Y(z)} \prod_{i=1}^{n} \Psi_i(y'_i, y'_{i-1}, z) \tag{2}$$

where $\Psi_i(y', y, z) = exp(W^T_{y',y} z_i + b_{y',y})$ are the potential functions and $W^T_{y',y}$ and $b_{y',y}$ are the weight vector and bias for the label pair (y', y) respectively. This binary potential or transition score is modeled as a matrix $[A]_{LXL}$. Here L is the number of possible labels. Each element in the matrix $A_{i,j}$ represents the transition score from label i to label j.

The logarithm of the conditional likelihood estimation is given by:

$$L(\boldsymbol{W}, \boldsymbol{b}) = \sum_i \log p(y \mid z; \boldsymbol{W}, \boldsymbol{b}) \tag{3}$$

The model is then trained end-to-end by maximizing the log-likelihood thereby choosing the parameters $L(\boldsymbol{W}, \boldsymbol{b})$.

2.5 LSTM-CRF Networks

Understanding the contribution of a LSTM network ([7]) to a CRF layer is critical to model further belief propogations for a given sentence. A LSTM network outputs a matrix of state transition scores $f_\theta([x]^T_1)$ for the sentence $[x]^T_1$ with parameters θ and for the

i-th tag and t-th word. A transition score matrix $[A]$ is generated by the LSTM network, with $[A]_i^j$ as the transition score from the i-th state to the j-th state for a pair of time instances. The LSTM is trained to maximize the log-liklihood with respect to θ as $\hat{\theta} \leftarrow \theta + \lambda \frac{\delta \log p(y|x,\theta)}{\delta \theta}$ using a gradient descent, where λ is the learning rate chosen as a Hyperparameter. These new parameters of the network $\hat{\theta}$ is updated as $\hat{\theta} = \theta \bigcup [A]_i^j \forall i, j$. The sentence score is the conditional tag probability of one tag path $[x]_1^T$ is given as :

$$s([x]_1^T, [i]_1^T, \theta) = \sum_{t=1}^{T} ([A]_{[i_{t-1}],[i]_t} + [f_\theta]_{[i]_t,t}) \quad (4)$$

We can write the conditional tag path probability for a sentence-level log likelihood as:

$$p(i \mid x, \theta) = \frac{\exp^{s([x]_1^T,[i]_1^T,\theta)}}{\sum_j \exp^{s([x]_1^T,[j]_1^T,\theta)}} \quad (5)$$

We now maximize the log sentence-level likelihood of the true path using $\hat{\theta}$, which can be given as the log of the conditional probability as :

$$\log p([y]_1^T \mid [x]_1^T, \hat{\theta}) = s([x]_1^T, [i]_1^T, \theta) - \underset{\forall j_1^T}{logadds}([x]_1^T, [j]_1^T, \hat{\theta}) \quad (6)$$

Viterbi algorithm can be used to find the maximum scored path and the parameters for it.

2.6 Bi-LSTM-CRF Networks

The Bidirectional LSTM-CRF model is mentioned in [8]. In a Bidirectional LSTM networks, for a given sentence, the network computes both a left $\overrightarrow{h(t)}$ and a right $\overleftarrow{h(t)}$ for a given sentence context in every input x(t). The final output is a result of concatenation h(t) = $[\overrightarrow{h(t)};\overleftarrow{h(t)}]$. The features in the layer h(t) are then used as input in a linear chain CRF interface to provide sequential decoding.

2.7 Bi-Directional Recurrent Neural Network with Attention (RNNA)

Since ADR relationships are not necessarily constrained to sentence level input, identifying the sentences or phrases that contribute to the identification of a ADR could be beneficial. A Bi-Directional RNN is used to encode the source document, the output of which is then input to a attention layer which generates the attention weights. The advantage of using attention mechanism is to figure out which encoded elements contributed to the generation of the current unit or the prediction of a ADR. We describe the attention mechanism as used with a Bi-Directional RNN in our model as represented with an *Encoder* and *Decoder* model.

2.7.1 Encoder: Bi-Directional RNN For Sequence Tagging. In Bidirectional LSTM networks, for a given sentence $(x_1 \ldots x_{T_x})$, the network computes a sequence of Forward hidden states $(\overrightarrow{h_1} \ldots \overrightarrow{h_{T_x}})$. A backward RNN reads the sequence in reverse order $(x_{T_x} \ldots x_1)$ and computes the backward hidden states$(\overleftarrow{h_1} \ldots \overleftarrow{h_{T_x}})$. The annotation for each word x_j is obtained by concatenating the forward and backward hidden states i.e. $h(t) = \left[\overrightarrow{h_j}\,;\,\overleftarrow{h_j}\right]$. The annotation h_j contains information of the preceding and following words for x_j. The sequence of annotations will be input to the decoder and the alignment model to compute the context vector.

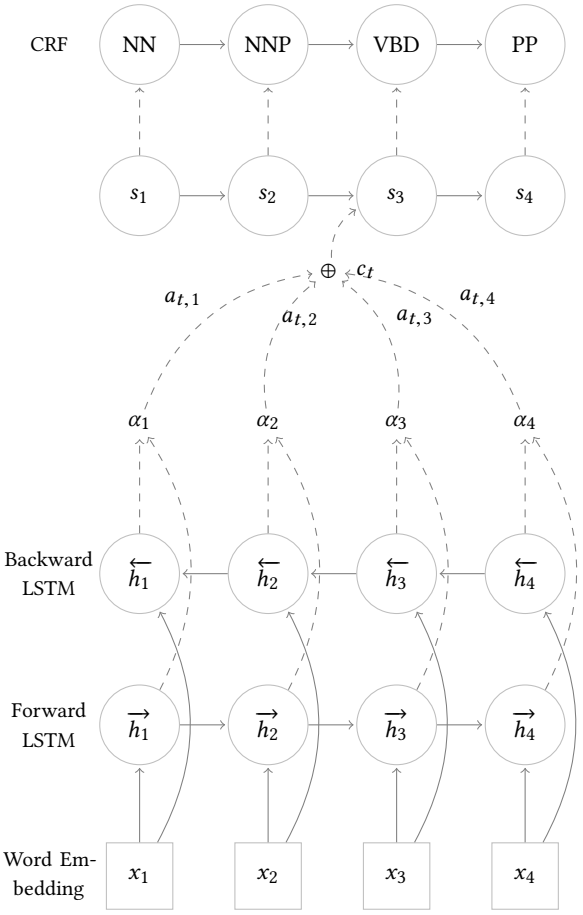

Figure 1: A Bi-Directional LSTM-CRF network with a forward and backward LSTM layer. The CRF layer receives input from the underlying hidden layers and then computes the unary potential from the parameters input.

2.7.2 Decoder: Attention Weights. Assume the encoding sequence of *annotations* output by the Bi-LSTM layers is $(h_1 \ldots . h_{T_x})$ for input sentence $(x_1 \ldots x_{T_x})$. The context vector c_i is computes as the weighted sum of annotations h_j:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (7)$$

The weight α_{ij} for each annotation is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (8)$$

where

$$e_{ij} = a(s_{i-1}, h_j) \quad (9)$$

Here a is the alignment model which scores how well the inputs around position j are modeled with the output at position i. The

energy e_{ij} reflects the importance of the annotation h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and the output y_i. The alignment model a as a feedforward neural network is jointly trained with other parameters such as the weight matrices. The alignment model computes a gradient of the cost function for backpropogation. The gradient can be used to train the alignment model. The context vectors align themselves to the target contexts which in the case of biomedical concepts can be relationships such as $Advice$, $Effect$, $Mechanism$ etc.

$$a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \quad (10)$$

Here v_a, W_a, U_a are the weight matrices. The context vector c_i is calculcated at every annotation step. The importance of a word as the similarity to a context is given by the attention weight in equation . We embed the biomedical concepts as a target word embedding matrix that can be used to align the context vectors with.

3 OUR MODEL

3.1 Bi-Directional RNNA with CRF

We apply Skip-Chain CRF, on top of the Bi-LSTM network to jointly model the probability of the entire tag sequence score. Unlike Linear CRF, Skip-Chain CRF can use the long term dependency between tags through the use of skip edges. In stead of taking into account the joint transition probability between every adjacent node, we can only consider the state transition between the words with attention. Linking the CRF graph edges based on the context vectors derived at each hidden state will classify the sequence with the presence of an ADR more appropriately.

4 EXPERIMENTS

Our extraction of disease, drugs and adverse effects from more than 6000 annotated electronic health records are in the form of a simple dictionary which needs to be embedded in a common space for representation with other word vectors. Our approach is to use the encoding of these concepts as found in Unified Medical Language System (UMLS) , LOINC and ICD-9, ICD-10 and NCD drug code and embed them in the common space of vectors using word2vec. We initialize the embedding layer at the start of the training with word vectors calculated on the larger data corpus. This ensures that words which are not seen frequently in the labeled data corpus still have a reasonable vector representation.

For training the network, a batch size of 100 is used, which means sentences whose total length is less than 100 are considered as a batch. We first create the word embeddings of dimension d and then each batch is further tokenized as words and each word is mapped to a real-valued vector form the word embedding. For each batch, we run the Bidirectional LSTM-CRF model which comprises of a forward pass of the hidden states $\overrightarrow{h(t)}$ and then a backward pass $\overleftarrow{h(t)}$ in a similar manner. The output if concatenated as $h(t) = [\overrightarrow{h(t)}; \overleftarrow{h(t)}]$ from which we get the output score for all tags at all positions. We then input the scores to calculate the attention weights for the words. On top of the Bi-LSTM, we use a CRF layer to compute the gradients of the network output and state transition edges.

We use ten-fold cross validation for the validating the performance, where 10% of the data is used as development set, 10% as the test set and the remainder for training. We use CRF-suite (Okazaki, 2007) for implementing the CRF tagger and Keras library to set up the neural network.

4.1 Hyperparameters

We choose a hidden layer size of 250 nodes for each of the forward and backward layers which is considered not too large or small for the experiment. The CRF layer has a hidden size of 200 nodes. The batch size of 64 sentences is chosen as with larger size the time taken to learn the parameters will be higher. The sentence length is restricted to 100 tokens. The first layer was a 200 dimensional word embedding layer. We used dropout with a probability of 0.5 for all models. All the models were trained with learning rate of 0.01, using Adagrad ([3]) with momentum.

4.2 Datasets

We use ADE corpus which was created by (Gurulingappa et al., 2012) by sampling from MEDLINE case reports. The case reports consists of signs, symptoms, diagnosis, treatment and follow-up for patients. The ADE corpus contains 2,972 documents with 20,967 sentences. We also use the ADE corpus from 1644 PubMed abstracts (Gurulingappa et al., 2012). The corpus was divided into datasets with ADE sentences and containing no ADEs.

Another dataset we use is the EHR documents in Case Record Interactive Search (CRIS) which was developed by South London and Maudsley (SLAM) NHS Foundation Trust with National Institute for Health Research (NIHR) Biomedical Research Centre (BRC) Infrastructure funding. It is an internal database of electronic health records for psychiatric patients. We have manually annotated 6000 EHR documents for presence of ADR and use these Gold annotated EHR documents for our experiment.

5 RESULTS

We compare the Baseline models of Bi-LSTM network only, Bi-LSTM with Linear chain CRF and Skip-Chain CRF networks. Precision, Recall and F-scores are calculated for positive extraction of ADR. It can be seen from the table shown that our model RNNA-CRF, which uses attention weights on the words alongwith a Skip-Chain CRF, has an improved performance over the baseline models when used with EHR documents. This maybe because the ADR description maybe spread over a document and contained in more than one sentences. Using attention weights as a parameter for Skip-Chain CRFs can relate these words for better prediction of ADR. All the neural networks constructed with Bi-LSTM model render similar performance results on PubMed articles. Our model RNNA-CRF has a very close recall (0.83) to Bi-LSTM-CRF network (0.80). We can visualize the words which are used by the neural network for predicting the labels with the help of visualizers for attention weights which can present results in more understandable form.

6 DISCUSSIONS

The work demonstrates the importance of creating custom word embedding from clinical concepts, as well as a modification of Recurrent neural networks to undertsand the contribution of words

Dataset	Model	Recall	Precision	F-score
MedLine	Bi-LSTM	0.8402	0.8720	0.8558
	Bi-LSTM-CRF	0.8068	0.8839	0.8436
	RNNA-CRF	**0.8523**	0.8917	**0.8716**
CRIS EHR	Bi-LSTM	0.7830	0.7845	0.7837
	Bi-LSTM-CRF	0.8021	0.8278	0.8147
	RNNA-CRF	**0.8316**	0.8222	**0.8268**

Table 1: Performance result comparison for Adverse Drug Reaction identification from two datasets.

and phrases in label prediction tasks. The models built on Bidirectional LSTM neural network alongwith CRF are both good in NER and relation extraction tasks. CRF on Bi-LSTM brings about improvement as compared to using a softmax layer or max-pooling. In the case of biomedical domains, incorporating our dictionary of Diseases, Drugs, Side-Effects, Negations into word-embeddings boosts the performance for EHR text documents. We used the PubMed Central Open Access Subnet (PMC) and PubMed word2vec embeddings. PMC is an online archive of over a million biomedical and life-sciences articles and the PubMed database has more than 25 million citations that cover abstracts of articles. Similarly incorporating known ADRs from SIDER2 ([11]), can help perform a direct lookup for drugs with known ADR mentioned. SIDER2 contains unstructured ADR data that can be mapped with the UMLS concept ID.

We note that as compared to identification of labels such as *Drug*, *Disease*, *Dosage*, *Severity*, *Frequency* are less complicated than the extraction of *ADR*. As the rules for labeling an ADR is not fixed and a ADR also taked into account text other than a *drug* or *disease* mention, there is a need to establish robust decoding algorithms for CRF.

There is certainly alot of future work to be done in this, for instance understanding which features are important for producing the context vectors and using those features in a Skip-Chain CRF. It is complicated to identify the correct features in biomedical domain to link on *skip-edges* of a CRF. Owing to the sparsity of labels in most EHR, attention weights should be considered as a useful resource in sequence labeling tasks.

ACKNOWLEDGMENTS

This research was funded by UCL BRC and supported by researchers at the National Institute for Health Research University College London Hospitals Biomedical Research Centre, and by awards establishing the Farr Institute of Health Informatics Research at UCLPartners, from the Medical Research Council, Arthritis Research UK, British Heart Foundation, Cancer Research UK, Chief Scientist Office, Economic and Social Research Council, Engineering and Physical Sciences Research Council, National Institute for Health Research, National Institute for Social Care and Health Research, and Wellcome Trust (grant MR/K006584/1)

REFERENCES

[1] A. R. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp* (2001), 17–21. http://view.ncbi.nlm.nih.gov/pubmed/11825149

[2] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: First results. *arXiv preprint arXiv:1412.1602* (2014).

[3] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[4] C. Friedman. 2000. A broad-coverage natural language processing system. *Proceedings of the AMIA Symposium* (2000), 270–4.

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

[6] Aron Henriksson, Maria Kvist, Hercules Dalianis, and Martin Duneld. 2015. Identifying adverse drug event information in clinical notes with distributional semantic representations of context. *Journal of biomedical informatics* 57 (2015), 333–349.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR* abs/1508.01991 (2015). http://arxiv.org/abs/1508.01991

[9] Ehtesham Iqbal, Robbie Mallah, Richard George Jackson, Michael Ball, Zina M Ibrahim, Matthew Broadbent, Olubanke Dzahini, Robert Stewart, Caroline, Johnston, and Richard JB Dobson. 2015. Identification of Adverse Drug Events from Free Text Electronic Patient Records and Information in a Large Mental Health Case Register. *PLoS One* 10, 8 (2015), e0134208.

[10] Abhyuday N Jagannatha and Hong Yu. 2016. Structured prediction models for RNN based sequence labeling in clinical text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, Vol. 2016. NIH Public Access, 856.

[11] Michael Kuhn, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2010. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology* 6, 1 (2010), 343.

[12] John Lafferty, Andrew McCallum, Fernando Pereira, and others. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, Vol. 1. 282–289.

[13] Chen Li, Runqing Song, Maria Liakata, Andreas Vlachos, Stephanie Seneff, and Xiangrong Zhang. 2015. Using word embedding for bio-event extraction. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015). Stroudsburg, PA: Association for Computational Linguistics*. 121–126.

[14] Christopher Longhurst, Robert Harrington, and Nigam Shah. 2014. A Green Button For Using Aggregate Patient Data At The Point Of Care. *Health Affairs* 33, 7 (2014), 1229–1235.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[16] Yifan Nie, Wenge Rong, Yiyuan Zhang, Yuanxin Ouyang, and Zhang Xiong. 2015. Embedding assisted prediction architecture for event trigger identification. *Journal of bioinformatics and computational biology* 13, 03 (2015), 1541001.

[17] Azadeh Nikfarjam, Abeed Sarker, Karen O'Connor, 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* (2015), ocu041.

[18] Naoaki Okazaki. 2007. CRFsuite: a fast implementation of conditional random fields (CRFs). (2007).

[19] G.K. Savova, J.J. Masanz, P.V. Ogren, J. Zheng, S. Sohn, K.C. Kipper-Schuler, and C.G. Chute. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association* 17, 5 (2010), 507–513.