Proceedings of Machine Learning Research 90:48–56, 2018 Medication and Adverse Drug Event Detection Workshop

Bidirectional LSTM-CRF for Adverse Drug Event Tagging in Electronic Health Records

Susmitha Wunnava * SWUNNAVA@WPI.COM
Xiao Qin † XQIN@WPI.COM
Tabassum Kakar ‡ TKAKAR@WPI.COM
Elke A. Rundensteiner RUNDENST@CS.WPI.EDU
Xiangnan Kong XKONG@WPI.COM
Worcester Polytechnic Institute, Worcester MA 01609, USA

Editor: Feifan Liu, Abhyuday Jagannatha, Hong Yu

Abstract

Adverse drug event (ADE) detection is a vital step towards effective pharmacovigilance and prevention of future incidents caused by potentially harmful ADEs. Electronic health records (EHRs) of patients in hospitals contain valuable information regarding the ADEs and hence are an important source for detecting ADE signals. We have developed a deep learning based system that utilizes a three layered deep learning architecture of 1) RNN (bi-directional long short-term memory (bi-LSTM)) for character-level word representation 2) bi-LSTM for context representation and 3) Conditional Random Fields (CRF) for the final output prediction, by integrating them into one deep network architecture. Furthermore, we have developed customized rule-based tokenization techniques for preprocessing text to deal with the noise in the EHR text. In this paper, we share our system architecture and its performance w.r.t the MADE1.0 NLP challenge.

Keywords: Adverse Drug Events, Named Entity Recognition, Deep Learning, and Natural Language Processing.

1. Introduction

Drug-related adverse events (ADEs) are known to be a leading cause of death in the United States. Early detection of the ADE incidents aids in the timely assessment, mitigation and prevention of future occurrences of severe, potentially fatal ADEs. Natural Language Processing (NLP) techniques towards recognizing ADEs and related information from spontaneous reports, clinical reports, electronic health records (EHR) provides an effective way of drug safety monitoring and pharmacovigilance.

A major challenge with processing EHR records is that EHR notes, while containing valuable knowledge correspond to unstructured text. Numerous challenges arise when ex-

* Susmitha Wunnava is thankful to the Seeds of STEM and Institute of Education Sciences, U.S. Department of Education for supporting her PhD studies via the grant R305A150571.
† Xiao is grateful to Oak Ridge Associated Universities (ORAU) for granting him an ORISE Fellowship to conduct research with the U.S. Food and Drug Administration.
‡ Tabassum is grateful to Oak Ridge Associated Universities (ORAU) for granting her an ORISE Fellowship to conduct research with the U.S. Food and Drug Administration.

tracting entities from such narratives. Often the notes contain medical and non-medical abbreviations, acronyms, numbers and misspelled words which make it difficult to recognize the critical information in the notes. In other words, certain types of information such as ADEs, Indications, Signs & Symptoms are harder to detect than others such as Drugname. This can be explained by the following. First, these entities can span across multiple words, about one to seven words per entity. Also, some entities could be expressed as a combination of entity-specific medical terms as well as non-medical descriptive text (Wunnava et al., 2018). For instance, in the phrase "*coronary artery disease related event prophylaxis*", the words "related" and "event" are descriptive text while the rest are medical terms. Moreover, there is a lot of ambiguity among relevant named entities. Depending upon the context, the same exact phrase can be either an ADE, indication or a sign & symptom (SSLIF). Table 1 states key challenges of textual notes. The example text is taken from the de-identified dataset of EHR notes of 21 cancer patients from the University of Massachusetts Medical School.

Table 1: Examples showing key challenges of biomedical text.

Challenges	**Example text**
Multiple words	*Lymphoplasmacytoid lymphoma involving bone marrow and spleen*
Medical and non-medical words	*cervix again is significantly stenotic*
Abbreviations	*IgG kappa monoclonal protein*
Ambiguous Named Entities	*Headaches - Indication or ADE or Sign or Symptom*

2. The MADE1.0 NLP Challenge

This section provides a brief introduction to the MADE1.0 NLP Challenges for Detecting Medication and Adverse Drug Events from Electronic Health Records hosted by University of Massachusetts at Lowell, Worcester, and Amherst[1]. The main objective of the challenge is to advance ADE detection techniques to improve patient safety and health care quality. The challenge consists of the following three tasks: 1) Named entity recognition (NER), 2) Relation identification (RI) and, 3) Integrated task (IT).

2.1. The Task

We have developed our system, DLADE (Dual-Level Embeddings for Adverse Drug Event Detection), specifically for *Task 1, the Named Entity Recognition (NER) problem* of the challenge. The task is to develop a system capable of automatically detecting any mentions of medication names and their attributes (dosage, frequency, route, duration) as well as mentions of ADEs, indications, other signs & symptoms. Tasks 2 & 3 (RI & IT) are beyond the scope of this paper.

1. https://bio-nlp.org/index.php/projects/39-nlp-challenges

2.2. Data Set

The MADE1.0 challenge used a total of 1089 de-identified EHR notes from 21 cancer patients. The notes are annotated with medication information (such as medication name, dosage, route, frequency, duration), adverse drug events (ADEs), indications and other signs and symptoms. The annotated notes were released in the BioC format (Comeau et al., 2013). 876 of these reports were released to participants of the competition for developing their learning system along with the gold standard annotation.

2.3. Resources

This challenge restricted the usage of tools and embeddings to the basic ones, such as NLTK, Stanford NLP, cTakes which should only be used for preprocessing text, in order to assure fairness among competition participants who included both university as well as company contributors with diverse resource access. The term *standard resources* refers to the training data released to the participating teams, the pre-trained word embedding trained using wiki, and de-identified Pittsburgh EHR and PubMed articles (Jagannatha and Yu, 2016a,b) and Unified Medical Language System (UMLS). The term *extended resources* refers to publicly available tools designed to work with medical concepts and medical relations as well as any ancillary corpus in addition to the standard resources. Our system, DLADE, is developed using only the training data and the pre-trained word embeddings released as part of the challenge.

2.4. Evaluation Process for MADE 1.0 Challenge

The developed system was then evaluated by the MADE1.0 organizers on two different tracks: 1) Standard track using only the standard MADE1.0 resources and, 2) Extended track using customized resources available publicly. The top teams for the AMIA 2018 Informatics Summit panel presentation were selected based only on the performance of each team for the Standard track. The evaluation is based on the strict matching in F1-score using exact phrase-level evaluation. Relaxed matching using word-level evaluation is not considered. The metrics used for evaluating the systems are Precision, Recall and F1-score. The best score is determined by the micro-averaged F1-score for the Standard track using an exact phrase-level evaluation. This simplified way selected a winner for this task of the competition.

3. The DLADE System

3.1. Preprocessing

As we will explain in Section 4, our model considers the EHR notes as a set of sentences where each individual sentence in turn consists of a sequence of words. Therefore, we first tokenize the EHR notes into sentences and then tokenize the words within each sentence. MADE 1.0 EHR notes contain noise (Figure 1), e.g, section headings with repeating punctuations and abnormal text formatting, e.g, unexpected line breaks where existing off-the-shelf tokenizers such as NLTK (Bird et al., 2009) fail to produce promising results. For this reason, we instead built a rule-based tokenizer that processes the EHR note character

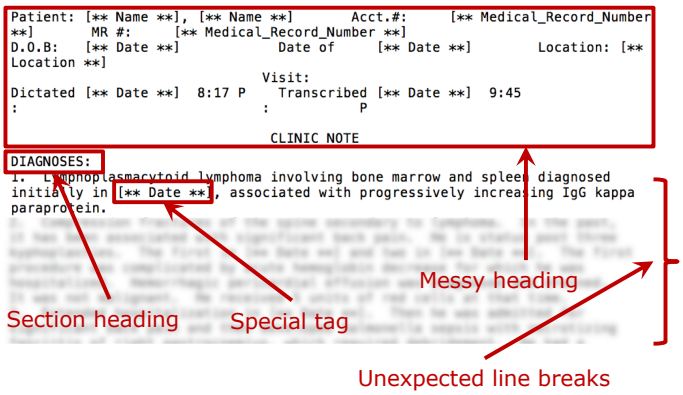

Figure 1: Noise in the EHR text.

by character for sentence and word chunking while concurrently recording the character offsets w.r.t the original text file.

Some named entities corresponds to multiple words. Hence we use the IOB (Ramshaw and Marcus, 1999) tagging scheme to distinguish between the beginning of an entity (tag B-*named entity*), or the inside of an entity (tag I-*named entity*). The no-entity tag is O.

3.2. Methods

In recent years, deep learning models especially Recurrent Neural Network (RNN) models have been shown to be promising techniques for sequence tagging and named entity recognition tasks due to their ability to learn from the context surrounding the words in a sequence (Lipton, 2015). Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a type of RNN that is effective at learning the long term dependencies between words in a sequence. Conditional Random Fields (CRF) (Lafferty et al., 2001) are probabilistic graphical models that have been used for sequence labeling tasks due to their ability to model the dependencies in the outputs of a sequence.

LSTM and CRF have their own advantages and disadvantages. LSTM is better for modeling long sequences of words, but the label for each word is predicted independently and not as a part of the sequence. CRF is better for modeling the entire sequence jointly, but need hand crafted features to obtain significantly good results. A combination of RNN and CRF models have also been explored and found to be effective for sequence tagging (Jagannatha and Yu, 2016b; Tutubalina and Nikolenko, 2017; Huang et al., 2015).

4. Our Model

Given the success of deep learning models for NLP tasks (Ma and Hovy, 2016a; Jagannatha and Yu, 2016b), we have developed a deep learning based system that utilizes the combined effectiveness of RNNs, more precisely bi-directional long short-term memory (bi-LSTM) (Schuster and Paliwal, 1997) models and CRF by integrating them into one deep network architecture. The bi-LSTM networks have been widely used for NLP tasks to learn the context representation of a word in a sequence by traversing through the sequence in both forward and backward (i.e. reverse order) directions.

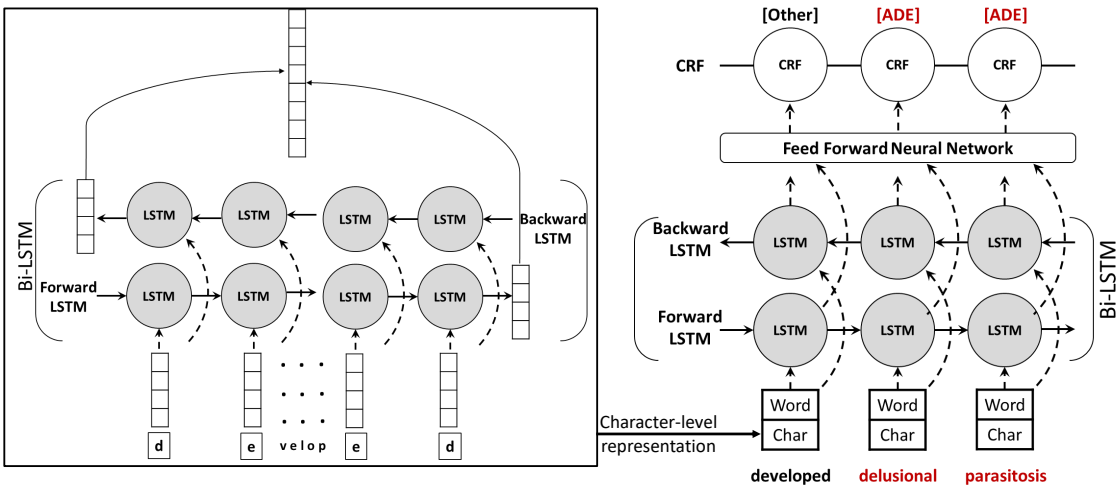

Figure 2: DLADE System Architecture

In a nutshell, our model is composed of a bi-LSTM neural network for an input layer responsible for character embedding, a second bi-LSTM for word embedding followed by a linear-chain CRF output layer. We have used the pre-trained medical word embeddings provided by the MADE1.0 challenge (Jagannatha and Yu, 2016a,b). More precisely, first at the bottom, character-level representations which capture the morphology of a word are computed by running a bidirectional-LSTM over the sequence of characters in the input words. A consolidated dense embedding, comprised of pre-trained medical word embedding concatenated with a learned character-level representation, is used to represent a word. Figure 2 shows our system architecture.

We feed this dense embedding of each word into a second bidirectional-LSTM. This second bi-LSTM then extracts the contextual representation of each word in the sentence that captures information from the meaning of the word, its characters and its context.

The output from the bidirectional-LSTM is used as input to a feed-forward neural network to compute a vector of scores, where each entry corresponds to a score for each tag. Tags are the individual named entities. To make the final prediction, the output of the feed-forward network is passed to a liner-chain CRF. The overall model is trained by minimizing the negative log-likelihood.

5. Experimental Results

5.1. Hyperparameter Settings

The named entities are Drug, Indication, Frequency, Severity, Dose, Duration, Route, ADE, SSLIF (other sign, symptom or disease). The model operates on the tokenized sentences. We used a batch size of 20 sentences. We did not make any restrictions on the sentence length. Rather, we used the maximum length of the sentences in a batch. All shorter sentences in that batch are padded with masks. As input, the pre-trained word embeddings

are 200 dimensional vectors and the learned character-level embeddings are 100 dimensional vectors. The hidden state is set to 100 dimensions for running bi-LSTM for learning character embeddings. The hidden state is set to 300 dimensions for running bi-LSTM with dense word embeddings. To avoid overfitting, we apply a dropout strategy (Srivastava et al., 2014; Ma and Hovy, 2016b) of 0.5 for our model. All the models were trained with learning rate of 0.001 using Adam (Kingma and Ba, 2014). Our models are trained on Intel(R) Xeon(R) 2.10GHz with a total memory of 251GB. They are implemented using the Tensorflow framework (Abadi et al., 2016).

Table 2: Evaluation Results on the Final MADE 1.0 Holdout Test Set.

	ADE	Dose	Drug	Duration	Frequency	Indication	Route	Severity	SSLIF	Micro-Avg
Precision	0.7261	0.8721	0.9066	0.7143	0.8438	0.6587	0.9100	0.7798	0.8309	**0.8373**
Recall	0.5644	0.8874	0.9019	0.8271	0.8412	0.6216	0.9381	0.8362	0.8570	**0.8454**
F1-Score	0.6351	0.8797	0.9042	0.7666	0.8425	0.6396	0.9239	0.8070	0.8438	**0.8413**

5.2. Methodology

Our system DLADE is trained on the 876 EHR notes from MADE 1.0. From the training set of sentences, 10% of the sentences are held out as validation set. This allows us to evaluate the model in the training phase by determining the best F1-score for early stopping. If there is no improvement in the f1-score within the last three consecutive epochs, the systems performs an early stopping.

5.3. Results on MADE1.0 Test Data Set

On the evaluation test set consisting of 213 EHR notes, our deep network achieves a micro-averaged precision, recall and F1-score of 0.8373, 0.8454,and 0.8413, respectively for the exact phrase-level evaluation. Table 2 shows our evaluation results on the MADE 1.0 evaluation test set for each of the entities. Our system has been selected as one of the top three performers and, is ranked first in the MADE 1.0 challenge for the Standard NER task.

To demonstrate the effectiveness of utilizing dual-level embeddings, we compare the prediction results from DLADE, which uses both the learned character-level representations of a word and the pre-trained word-level embeddings with a baseline system that utilizes only the pre-trained word-level embeddings.

Table 3 compares the F1-scores of individual entities as well as the overall micro-averaged F1-score of all entities combined. It shows the percentage improvement with DLADE using dual-level embeddings over the baseline system using only word embeddings. We use the pairwise t-test on the F1-score to calculate the statistical significance of our scores. The improvement in F1-score for DLADE as compared to our baseline is statistically significant ($p < 0.05$ and $p < 0.01$). Of all the entities, *Duration* showed a large improvement (11.4%) from utilizing the dual-level embeddings. Duration labels are challenging to detect because they often are comprised of phrases that contain non-medical text and contain numbers such as, “four cycles”, “14 days”, “day 1 through 14”, “over 15 minutes”,“two weeks”. They can be easily misclassified and treated as the *Outside* or no-entity tag O.

Table 3: Improvement for MADE 1.0 in F1-score when using Dual-Level Embeddings.

	Word Embedding	Dual-Level (Character + Word) Embedding	Improvement
ADE	0.5848	0.6351	8.6%
Dose	0.8172	0.8797	7.6%
Drug	0.8780	0.9042	3.0%
Duration	0.6879	0.7666	11.4%
Frequency	0.7964	0.8425	5.8%
Indication	0.6151	0.6396	4.0%
Route	0.8705	0.9239	6.1%
Severity	0.7648	0.8070	5.5%
SSLIF	0.8290	0.8438	1.8%
Micro-Averaged	**0.8147**	**0.8413**	**3.3%**

6. Error Analysis of DLADE System

An error analysis was performed to understand the source of errors generated by the NER system. We inspected and evaluated instances for which the system incorrectly predicted the phrases, considering both false positive and false negative cases.

- One of the challenges as shown in Table 1 is that the entity can span across multiple words. In this case, it is critical to extract the phrase in its entirety to retain the true meaning of the phrase. For this example, our system was able to correctly extract the entire phrase "*nodular sclerosing Hodgkin disease involving the mediastinum and both necks*". This contains 10 words. However, the phrase was misclassified as *Indication* when it actually is an *SSLIF*.

- Another challenge is the mixture of medical and non-medical text in the entity phrase. This makes it difficult to detect the entity as a whole. For instance, the phrase "inflammation of your liver or gallbladder or your pancreas" was annotated as *SSLIF*. Although our system detected the phrase correctly as *SSLIF*, it missed the last two words "your pancreas" of the phrase. This meant that our result was labeled as *Other entity-O* wrongly even though it mostly was correct.

- The occurrence of medical abbreviations text is rare in the training set. Although our system was able to correctly detect certain entities that contain abbreviations such as "stage IIA"(*Severity*), "HPV"(*SSLIF*), there are few other entities with abbreviations such as "SIL cytology"*SSLIF* where our system failed to recognize the phrase and categorized it as a no-entity label *O*.

- Due to the ambiguous nature of Indication, ADE, and SSLIF entity words and phrases, it is very challenging to differentiate between these two types of labels. For example, in the two sentences: 1) "the back pain (*Indication*) started about 10 o'clock last night"

and, 2) "reports weight gain (*ADE*) and increased (*ADE*) appetite from corticosteroid therapy", our system misclassified the *Indication* and *ADE* labels as *SSLIF*.

7. Conclusion

In conclusion, we have shown that the integration of two widely used sequence labeling techniques that compliment each other along with dual-level embeddings (character-level and word-level) to represent words in the input layer results in a deep learning architecture that achieves excellent information extraction accuracy for EHR notes.

Acknowledgments

- We are grateful to Dr. Marni Hall (former Sr. Program Director), Suranjan De (Deputy Director), Sanjay K. Sahoo and Thang La, Regulatory Science, OSE, FDA for introducing us to Pharmacovigilance in general and the ADE detection problem in particular. We also thank WPI DSRG members for their valuable feedback.
- For this project, we have adapted open source code available from GitHub on sequence tagging with Tensorflow (Genthial, 2017).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009. ISBN 978-0-596-51649-9. URL http://www.oreilly.de/catalog/9780596516499/index.html.

Donald C Comeau, Rezarta Islamaj Doğan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, et al. Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, 2013, 2013.

Guillaume Genthial. Named entity recognition with tensorflow. *GitHub repository*, 2017.

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

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Abhyuday N Jagannatha and Hong Yu. Bidirectional rnn for medical event detection in electronic health records. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2016, page 473. NIH Public Access, 2016a.

Abhyuday N Jagannatha and Hong Yu. 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*, volume 2016, page 856. NIH Public Access, 2016b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015. URL http://arxiv.org/abs/1506.00019.

Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of ACL*, pages 1064–1074, 2016a.

Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016b. URL http://aclweb.org/anthology/P/P16/P16-1101.pdf.

Lance A Ramshaw and Mitchell P Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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

Susmitha Wunnava, Xiao Qin, Tabassum Kakar, Xiangnan Kong, Elke A. Rundensteiner, Sanjay K. Sahoo, and Suranjan De. One size does not fit all: An ensemble approach towards information extraction from adverse drug event narratives. In *Proceedings of HEALTHINF*, pages 176–188, 2018. ISBN 978-989-758-281-3.