IBM Research System at TAC 2017: Adverse Drug Reactions Extraction from Drug Labels

Bharath Dandala
IBM Research
Yorktown Heights, NY.
bdand@us.ibm.com

Diwakar Mahajan
IBM Research
Yorktown Heights, NY
dmahaja@us.ibm.com

Murthy Devarakonda
IBM Research
Yorktown Heights, NY.
mdev@us.ibm.com

Abstract

Identifying Adverse Drug Reactions (ADRs) is an important concern in clinical medicine. Automatic extraction of ADRs from the corpus enable many clinical decision support applications, and drug labels or 'package inserts' are among the primary sources of information. The Adverse Drug Reaction extraction from Drug Labels challenge at TAC 2017 defined several NLP tasks leading up to ADR extraction from a subset of drug labels. We participated in the Task 1 (extraction of mentions of severity, drug class, negation, animal, and factor) and the Task 2 (identification of relationship types - negated, hypothetical and effect - between ADRs and related assertions). For Task 1, we implemented a joint Bi-directional-LSTM (BiLSTM)-CRF and Attention-BiLSTM neural network for identifying contiguous, discontiguous and overlapping mentions simultaneously. For Task 2, we used another Attention-BiLSTM network. Our system achieved F-measures of 78.21 for untyped exact match and 78.00 for typed exact match, for Task 1. Using the mentions generated by our system, we achieved F-measures of 45.16 and 44.60 for the binary relation detection (Task 2) without and with type identification respectively. However, using the gold standard mentions we achieved 91.20 and 87.86 F-measures for the Task 2. Thus, through this work, we demonstrated effective adaptation of BiLSTM networks for a subset of ADR extraction tasks.

Index Terms: Deep Learning, Adverse Drug Reaction, Drug Labels, Named Entity Recognition, Relation Extraction, BiLSTM-CRF, Attention-BiLSTM

1 Introduction

Identifying Adverse Drug Reactions (ADR) is an important concern for patients, physicians, researchers, regulatory authorities, and drug manufacturers alike. However, collecting and maintaining an ADR repository is largely a tedious manual task. Usually, ADRs for a drug are identified in one of two phases. First, during Phase III of clinical trials, ADRs of a drug are carefully observed and the data is recorded. This data forms the 'Drug Label Data', which is added as 'package inserts' with the drugs. The US Food and Drug Administration (FDA) strictly regulates the content and format of this information. However, the drug label data can vary among various drug manufacturers for the same drug. The second phase of ADR detection for a drug takes place, once the drug is on the market. Physicians may observe and report ADRs to systems like FDA Adverse Event Reporting System (FAERS). Maintaining an accurate ADR repository not only requires consolidating the drug label data from various manufacturers but also reviewing the FAERS reports to determine if the reported post-marketing ADR is undetected or missing in the current drug label data. This process is largely manual and labor intensive. Thus, to improve the efficiency of this process, it is desirable to automate extraction of ADRs from drug labels.

Information Extraction methods for Named Entity Recognition (NER) and Relation Extraction is a fundamental requirement in automatic ADR extraction for drug labels. Accuracy of these foundational analytics will significantly impact ADR curation and further, has the potential to improve clinical decision support systems. TAC 2017 challenge on "Adverse Drug Reaction Extraction from Drug Label Data Challenge" identified multiple tasks involved in ADR extraction from the drug labels data. BiLSTM-CRF models (Huang et al., 2015) has previously shown to accurately recognize continuous entities in clincal and biomedical NLP data sets (Chalapathy et al., 2016; Habibi et al., 2017). Li et al., (2017) enhanced this method to recognize not only the contiguous entities but also non-contiguous and overlapping entities on biomedical data sets from ShARe/CLEF eHealth Evaluation Lab 2013 (Suominen et al., 2013) and GENIA v.3.02 (Kim et al., 2003). In this reserach, we adapted and improved upon Li et al., method to extract concepts with their semantic types, where the semantic types include Adverse Reaction, Animal, Drug Class, Factor, Negation, Severity and Effect. Further, we used Attention-Based BiLSTM networks for the relation classification task, where the relations include effect, negated, and hypothetical assertions. In this work, our contributions are as follows:

- State-of-the-art deep learning architectures for Named Entity Recognition and Relation Extraction for ADR extraction.
- Novel technique for identifying disjoint (discontinuous and overlapping) entities using Attention-BiLSTM.

The rest of the paper is organized as follows: in Section 2, we describe the dataset and label encoding schemes used in concept extraction for this challenge. In Section 3, we present our system architecture and methods for the concept and relation extraction tasks. In Section 4, we describe experimental settings of the system and achieved results for different settings and parameters. In Section 5, we conclude with our insights and details about the future direction.

2 Datasets and representation

2.1 Datasets

In a collaborative effort, National Library of Medicine(NLM) and U.S. Food and Drug Administration (FDA) manually annotated 101 drug label documents with concepts, relations and reactions. This data which consists of 15722 mentions (13795 Adverse Reactions, 44 Animal, 249 Drug Class, 602 Factor, 98 Negation, 934 Severity), 3228 relations (1454 Effect, 1611 Hypothetical and 163 Negated) and 7038 reactions. Furthermore, the organizers released an unannotated dataset comprises of 2208 drug label documents. Combined, these represent a significant core study set of labels of interest to the FDA. Furthermore, to evaluate the participant systems, a separate test dataset with 100 drug label documents is created and kept blind. This dataset consists 13735 mentions (12317 AdverseReactions, 27 Animal, 94 DrugClass, 520 Factor, 102 Negation 675 Severity) and 2039 relations (695 Effect, 1225 Hypothetical, 119 Negated).

2.2 Label encoding and decoding schemes

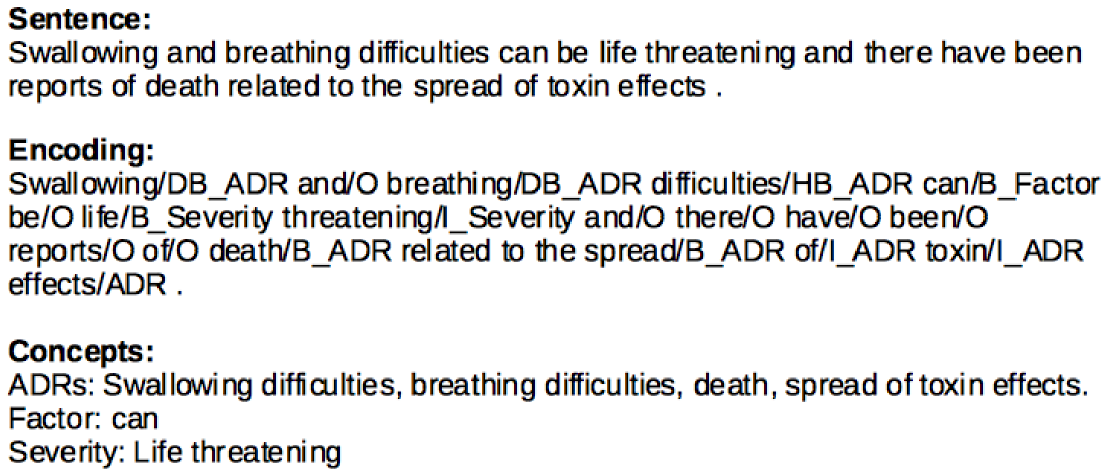
Typically, the named entities or concepts are continuous sequences of words. Thus, in machine learning-based named entity recognition (NER) systems, annotated data is encoded using BIO tagging, where each word is assigned into one of three labels: **B** means beginning, **I** means inside, and **O** means outside of a concept. However, BIO encoding is not sufficient for disjoint concepts. In the 101 manually annotated drug label documents about 7% (1078/15722) of mentions are disjoint concepts with overlapping words or discountinous spans.

With the advent of disjoint concepts in recent NER challenges, Tang et al., (2013; 2015) tried to address this problem by using alternative label encodings such as: BIOHD and BIOHD1234. Recently, Li et al. (2017) used BIOHD encoding and proposed a decoding scheme that is better suited for such an encoding. Furthermore, their system showed improvement on several NER datasets compared to using BIO tagging. Thus, in this paper, we adapted their encoding and decoding techniques that contains 7 labels {B I O HB HI DB DI} in which:

- HB and HI refers to tokens that are shared by multiple concepts. These words are the overlapped portions of disjoint concepts. We refer to these token/sequence of tokens as *head components*.
- DB and DI refers to tokens that belong to discontinuous concepts, however these tokens not shared by multiple concepts. We refer to these token/sequence of tokens as *discontinuous components*.
- B and I are used to label the tokens that belong to continuous concepts and,
- O refers to tokens that are outside of concepts.

Figure 1 shows an example with BIOHD label encoding with semantic type and annotated concepts.

During the decoding stage, given an input sentence, as a first step, we first identify continuous concepts and the components of disjoint concepts *(head and discontinuous components)*. Next our system predicts whether each pair of the extracted disjoint components should be combined or not. It is trivial to merge when only two such components are detected in the given sentence and it is not otherwise.



Sentence:
Swallowing and breathing difficulties can be life threatening and there have been reports of death related to the spread of toxin effects .

Encoding:
Swallowing/DB_ADR and/O breathing/DB_ADR difficulties/HB_ADR can/B_Factor be/O life/B_Severity threatening/I_Severity and/O there/O have/O been/O reports/O of/O death/B_ADR related to the spread/B_ADR of/I_ADR toxin/I_ADR effects/ADR .

Concepts:
ADRs: Swallowing difficulties, breathing difficulties, death, spread of toxin effects.
Factor: can
Severity: Life threatening

Figure 1: BIOHD encoding for concept extraction.

When more than two such components are present in a sentence, we construct a graph G = {V,E}, where the vertex set V represents all components and the edge set E represents the positive relations predicted from the second step. The decoding objective is to extract all the cliques in graph. Finally, all the components in a clique compose an integrated concept. For a complete understanding of this encoding and decoding schemes please refer to (Li et al., 2017).

3 Architectures of Concept and Relation Extraction

With the recent advancements in deep learning research, several neural network architectures have been successfully applied to concept and relation extraction. Among these, architectures based on bi-directional LSTMs are proven to be very effective (Huang et al., 2015; Ma and Hovy, 2016; Zhou et al., 2016; Zhang and Wang, 2015). In this section, we describe our concept and relation extraction systems in detail. The architectures of our concept and relation extraction systems are illustrated in Figure 2 and Figure 3 respectively.

3.1 Bi-directional LSTM

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (RNN) that models interdependencies in sequential data and addresses the vanishing or exploding gradients (Bengio et al., 1994) problem of vanilla RNNs by using adaptive gating mechanism.

Given a input sequence x=$(x_1, x_2...x_T)$ where T is the sequence length, LSTM hidden state at timestep t is computed by:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^i * x_t + \mathbf{U}^i * h_{t-1} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f * \mathbf{x}_t + \mathbf{U}^f * h_{t-1} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o * \mathbf{x}_t + \mathbf{U}^o * h_{t-1} + \mathbf{b}^o) \\
\mathbf{g}_t &= \tanh(\mathbf{W}^g * \mathbf{x}_t + \mathbf{U}^g * h_{t-1} + \mathbf{b}^g) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(c_t)
\end{aligned} \tag{1}$$

where $\sigma(.)$ and $\tanh(.)$ are the element-wise sigmoid and hyperbolic tangent functions, $\odot$ is the element-wise multiplication operator, and i_t, f_t, o_t are the input, forget and output gates. h_{t-1}, c_{t-1} are the hidden state and memory cell of previous timestep respectively.

Unidirectional LSTMs suffer from weakness of not utilizing the future contextual information. Bi-directional LSTM (Graves and Schmidhuber, 2005; Graves, 2013) addresses this by using two independent LSTMs (forward and backward) in which one processes the input sequence in the forward direction, while the other processes the input in the reverse direction. The forward LSTM computes the forward hidden states ($\overrightarrow{h_1}$, $\overrightarrow{h_2}$, ... $\overrightarrow{h_t}$) while the backward LSTM computes backward hidden states ($\overleftarrow{h_1}$, $\overleftarrow{h_2}$, ... $\overleftarrow{h_n}$) . Then for each timestep t , the hidden state of the Bi-LSTM is generated by concatenating $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$

$$\overleftrightarrow{h_t} = (\overrightarrow{h_t}, \overleftarrow{h_t}) \tag{2}$$

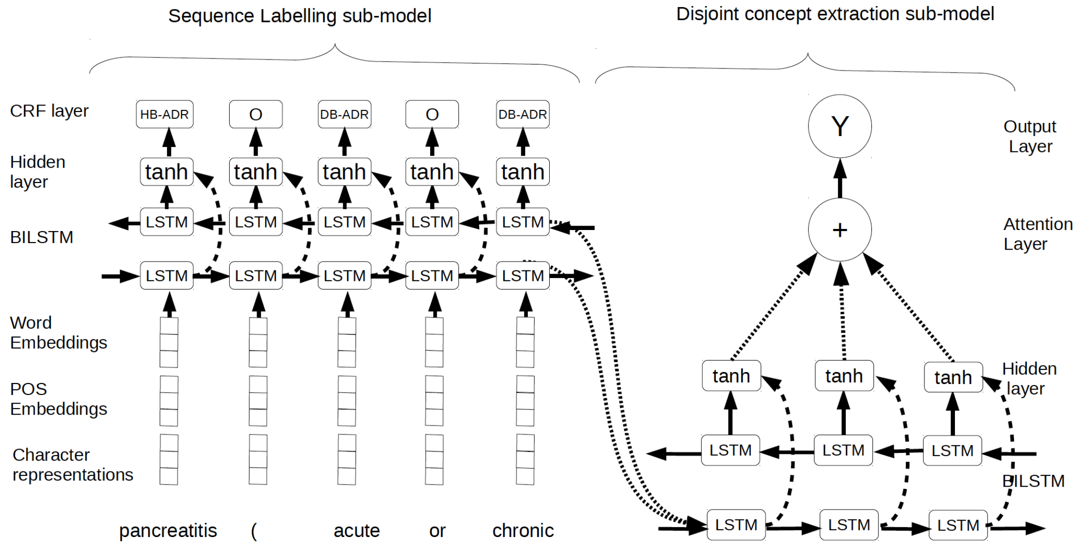


Figure 2: Architecture of Concept Extraction.

3.2 CRF Layer

Although Bi-directional LSTM networks have the ability to capture long distance inter-dependencies, previous research suggests additionally capturing the correlations between adjacent labels can help in sequence labeling problems (Lample et al., 2016; Collobert et al., 2011; Huang et al., 2015). Conditional random fields (CRF) (Sutton et al., 2012) helps in capturing these correlations between adjacent tags. Given an observation sequence $\overleftrightarrow{h_t}$ (outputs from Bi-directional LSTM), CRF jointly models the probability of the entire sequence of labels Y=$(y_1, y_2 ... y_T)$ by using the discriminative probability to y_i given x_i and the transition probability between adjacent labels.

3.3 Attention Bi-directional LSTM

Attention mechanism is a technique often used in neural translation of text introduced in (Bahdanau et al., 2014). The attention mechanism allows the networks to selectively focus on specific information. This benefited serveral natural language processing (NLP) tasks such as factoid question answering (Hermann et al., 2015), machine translation (Bahdanau et al., 2014) and relation classification(Zhou et al., 2016). In this paper, we used attention mechanism in disjoint entity recognition sub model of concept extraction task (see Figure 2) and relation classification task (see Figure 3) similar to (Zhou et al., 2016)

Formally, let H be a matrix consisting of output vectors $[\overleftrightarrow{h_1}, \overleftrightarrow{h_2} .... \overleftrightarrow{h_t}]$ (outputs from Bi-directional LSTM network), the representation r of the input is formed by a weighted sum of these output vectors:

$$\begin{aligned} \mathbf{M} &= \tanh(H) \\ \alpha &= softmax(w^T * H) \\ \mathbf{r} &= H * \alpha^T \end{aligned} \qquad (3)$$

where $H \varepsilon R^{d^w X T}$, d^w is the dimention of vectors, w^T is the transpose of trained parameter vector. We obtain the final representation from:

$$\mathbf{h}^* = tanh(r) \qquad (4)$$

3.4 Architecture of Concept Extraction

As shown in Figure 2, our concept extraction architecture contains two sub-models, namely, 1) Sequence Labelling sub-model and 2) Disjoint concept extraction sub-model.

Sequence Labelling sub-model: The *sequence labeling sub-model* labels each word of an input sentence. This sub-model contains embedding layer to which we feed word, part-of-speech embeddings and character representations. Next, it contains bi-directional long-short term memory layer (as introduced in 3.1) which takes inputs from embedding layer and transforms them to high level features or representations. Then, these features are fed into the CRF (as introduced in 3.2) for labeling each word as

Algorithm 1: Algorithm for concept extraction

```
foreach epoch do
    foreach batch do
        Sequence Labelling sub-model:
        1) forward pass for forward-state LSTM and backward-state LSTM
        2) forward and backward pass for CRF layer
        3) backward pass for forward-state LSTM and backward-state LSTM
        4) update parameters
        Disjoint concept extraction sub-model:
        5) forward pass for forward-state LSTM and backward-state LSTM
        6) attention mechanism
        7) backward pass for forward-state LSTM and backward-state LSTM in Disjoint concept
           extraction sub-model
        8) backward pass for forward-state LSTM and backward-state LSTM in Sequence Labelling
           sub-model
        9) update parameters
```

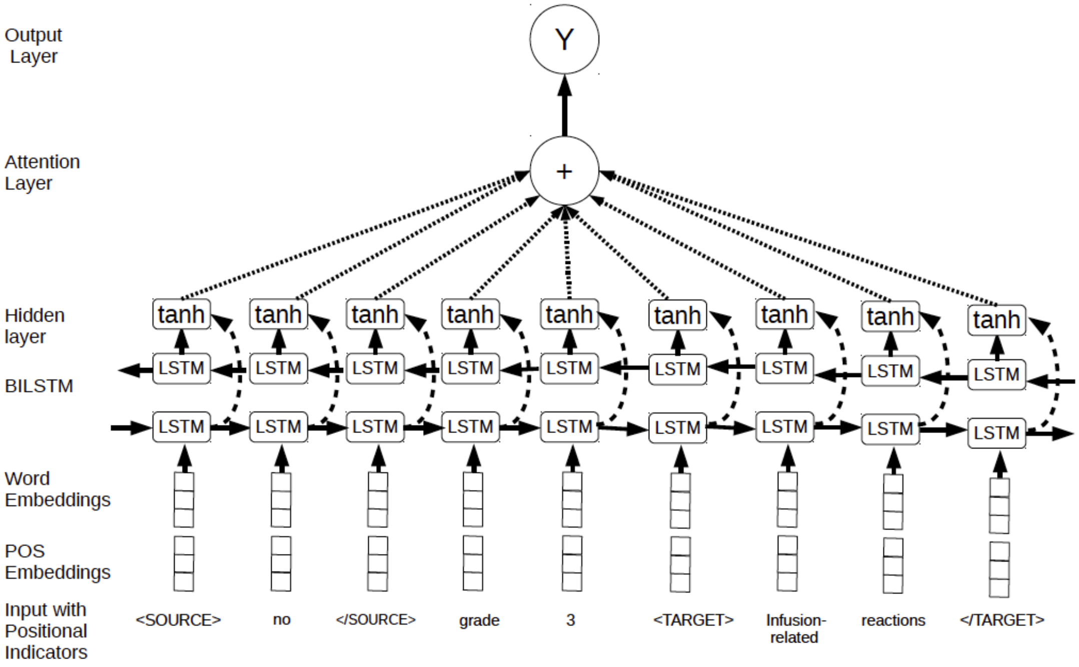


Figure 3: Architecture of Relation Extraction.

shown in Figure 2. Finally, we adopt the Viterbi algorithm for training the CRF layer and decoding the optimal output label sequence.

Disjoint concept extraction sub-model: As an alternative step, a subset of the features from BiLSTM layer *Sequence Labelling sub-model* (features for words between and including the words in each pair of discontinuous or header components) are fed into the BiLSTM layer of *Disjoint concept extraction sub-model*. Additionally, we insert positional indicators <TARGET></TARGET> around the target components. Consequently, outputs of this BiLSTM layer are fed into attention layer which produces a weight vector, and merges word-level features from each time step into context-level feature vector. Finally, a softmax function is used to determine whether the target components should be combined or not.

The training procedure for concept extraction is shown in Algorithm 1. We generate one training data representation for the *sequence Labelling sub-model* and another for the *disjoint concept extraction sub-model*. The former contains sequence of tokens from the sentences and corresponding label assigned according to the label scheme introduced in Section 2.2 and the latter representation contains sentences (only words between and in the target components) for each pair of components, positional indicators `<TARGET></TARGET>` around target components and corresponding label indicating whether these concepts are connected or not.

We exploit back propagation to update the parameters of these two sub models. Adam optimizer is employed to control the update step. In addition, we add L2 regularization and utilize dropout to alleviate the over-fitting problem. For each batch, in the training dataset, we train these sub-models alternatively. The parameters of LSTM units in the *sequence labeling sub-model* are shared by both sub-models, thus the loss of each batch can propagate and update these parameters.

3.5 Architecture of Relation Extraction

The architecture of our relation extraction system is illustrated in Figure 3. We used Attention-BiLSTM architecture introduced by (Zhou et al., 2016) for relation classification. This network contains:

Input layer: This layer takes tokens, part-of-speech tags from a sentence as input. Zang et. al(2015) first introduced the position features (PF) in bi-directional LSTM architures and demonstrated their use for relation classification task. These features are derived from the relative distances of the current token to the target pair of concepts. Further, Zhou et. al(2016) replaced positional features with much simpler positional indicators (as shown in Figure 3 for concepts “no” and “infusion-related reactions”) and demonstrated similar results. Thus, we used positional indicator tags around the tokens of target pair of concepts. Finally, in the dataset introduced in 2.1, the relationship is always between adverse reactions and other concept types. Thus we reserved <TARGET></TARGET> positional indicators for adverse reactions and <SOURCE></SOURCE> positional indicators for other concept types.

Embedding layer: The embeddings layer maps each token and its pos-tag into a low dimension vector.

Bi-directional LSTM layer: This layer transforms the inputs from the embedding layer to high level features;

Attention layer: This layer merges word-level features from each time step into a sentence-level feature vector using the attention mechanism introduced in section 3.3

Output layer: This layer takes sentence-level feature vector as input and uses a softmax classifier to classify the relation associated between target concepts.

4 Experiments and Results

4.1 Experimental Settings

Concept Extraction			
parameter	**BIO**	**NerOne**	**OurSystem**
dropout	0.4	0.4	0.5
learning rate	0.02	0.03	0.03
regularization	$1e^{-7}$	$1e^{-6}$	$1e^{-6}$
hidden layer	150	100	100

Table 1: Hyperparameters for concept extraction.

Relation Extraction		
parameter	**Extraction**	**Classification**
dropout	0.5	0.5
learning rate	0.01	0.01
regularization	$1e^{-7}$	$1e^{-5}$
hidden layer	100	100

Table 2: Hyperparameters for concept extraction.

Concept Extraction			
	Precision	**Recall**	**F1**
Exact (-type)	81.12	75.50	78.21
Exact (+type)	80.90	75.30	78.00
Relation Extraction			
Binary	54.62	38.50	45.16
Binary (+type)	53.94	38.02	44.60
Full (-type)	48.64	32.89	39.24
Full (+type)	48.13	32.54	38.83

Table 3: Results for concept and relation extraction.

We used 20% of the training data as our development set. We used Stanford CoreNLP toolkit(Manning et al., 2014) for tokenization, sentence segmentation and part-of-speech tagging. The

Relation Classification			
	Precision	**Recall**	**F1**
Binary	90.06	92.38	91.20
Binary (+type)	86.75	88.99	87.86
Full (-type)	87.48	85.03	86.24
Full (+type)	84.09	81.74	82.90

Table 4: Results for relation classification.

Concept Extraction			
	Precision	**Recall**	**F1**
BIO Tagging	81.10	70.30	75.31
NerOne	79.80	74.30	76.95
Our system	80.90	75.30	78.00

Table 5: Results for concept and relation extraction.

word and character embeddings are pre-trained, using word2vec (Mikolov et al., 2013). The training data for the embeddings is *unannotated adverse drug reaction documents* released as part of the task. We fixed word embedding length to 200, character embedding length to 50 and part-of-speech embedding length to 20. The part-of-speech embeddings are initialized randomly.

Hyperparameters: There are four hyperparameters in our models, namely the dropout rate, learning rate, regularization parameter, and hidden layer size. The hyperparameters for our models were tuned on the development set for each task. Previous research suggests using dropout mitigates over-fitting and especially beneficial to the NER task(Ma and Hovy, 2016). We experimented by tuning the hyperparameters with different settings: dropout rates (0.0, 0.1, 0.2, 0.3 and 0.4,0.5), hidden layer sizes (100,150,200) and regularization parameter ($1e^{-5}, 1e^{-6}, 1e^{-7}$,$1e^{-8}$.). We chose Adam (Kingma and Ba, 2014) as our stochastic optimizer and tuned the learning rate at (0.01,0.02,0.03). .We used early stopping(Graves, 2013) based on performance on development dataset. The best performance appear at around 20 epochs and 15 epochs for concept extraction and relation extraction respectively.

4.2 Results

Table 3 shows our submitted results on test dataset for both concept and relation extraction tasks. These results are obtained by using the hyperparameters shown in Table 1 and Table 2 for concepts and relation extraction tasks respectively. These hyperparameters are obtained by tuning them on development set. For the concept extraction task, we achieved F-measure of 78.21 for recognizing the concept spans and F-measure of 78.00 for recognizing the concept spans with their semantic types. Furthermore, we compared the test dataset results of the BiLSTM-CRF model using BIO tagging, NerOne (Li et al., 2017) with our system. For a fair comparision, we used parameter tuning in all these systems and used same input features/embeddings for all these systems. As shown in Table 5 , both NerOne and our system outperformed BiLSTM-CRF with BIO tagging. Moreover, our system achieved the highest precision and recall and outperformed state-of-art NerOne system for recognizing continous disjoint concepts.

For the relation extraction task, we achieved F-measure of 45.16 and 44.60 for determining binary relations without and with semantic type of target concepts respectively. Also, we achieved F-measure of 39.28 and 38.83 for determining relation type without and with semantic type of target concepts respectively.

We separately conducted experiments providing the gold-standard concepts to our relalation extraction system. The results are presented in Table 4 and we achieved significantly high F-measure of 82.90 compared to 38.83 with predicted concepts. This indicates further improvements is needed in our concept extract model.

5 Conclusion and Future Work

We reported on using state-of-the-art deep learning neural networks for identifying mentions and relations relevant to ADR extraction. We used a novel BiLSTM-CRF models for identifying contiguous mentions and Attention-BiLSTM for identifying discontiguous mentions and relation extraction. Accuracy of mentions identification and relation extraction using gold labels for mentions was high on the official test set (F measures were about 78.0 for mentions, and between 81.0 and 91.0 for relation classification). However, F measures for relation extraction with mentions that were detected by our analytics were low (only in the range of 38.0 to 45.0). Further analysis showed that the poor performance of our model for low frequency classes resulted in poor results in relation extraction, because these low frequency classes appeared in disproportionately large number of relations. In the future, we plan to address low frequency relation classes in more detail and separately. Also, we plan to exploit semi-structured nature (tables, lists, sections) of drug label data for both the tasks. Deep learning models proved to be effective in detecting adverse reactions (including the discontiguous mentions) and relations . Thus, we continue to pursue this research direction in more depth to further improve our system.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5(2):157–166.

Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. 2016. Bidirectional lstm-crf for clinical concept extraction. *arXiv preprint arXiv:1611.08373* .

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5):602–610.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 33(14):i37–i48.

Karl Moritz Hermann, Tomáš Kociskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340* .

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

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

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. Genia corpusa semantically annotated corpus for bio-textmining. *Bioinformatics* 19(suppl_1):i180–i182.

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

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* .

Fei Li, Meishan Zhang, Bo Tian, Bo Chen, Guohong Fu, and Donghong Ji. 2017. Recognizing irregular entities in biomedical text via deep neural networks. *Pattern Recognition Letters* .

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* .

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, et al. 2014. The stanford corenlp natural language processing toolkit. In *In ACL, System Demonstrations*. Citeseer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R South, Danielle L Mowery, Gareth JF Jones, et al. 2013. Overview of the share/clef ehealth evaluation lab 2013. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, pages 212–231.

Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4(4):267–373.

Buzhou Tang, Qingcai Chen, Xiaolong Wang, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, and Hua Xu. 2015. Recognizing disjoint clinical concepts in clinical text using machine learning-based methods. In *AMIA*. AMIA.

Buzhou Tang, Yonghui Wu, Min Jiang, Joshua C. Denny, and Hua Xu. 2013. Recognizing and encoding discorder concepts in clinical text using machine learning and vector space model. In *CLEF (Working Notes)*. CEUR-WS.org, volume 1179 of *CEUR Workshop Proceedings*.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006* .

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 207–212.