Extracting Adverse Drug Reactions using Deep Learning- and Dictionary-Based Approaches

Mert Tiftikci[1], Arzucan Özgür[1], Yongqun He[2], and Junguk Hur[3.$]

[1] Department of Computer Engineering, Bogazici University, Istanbul, Turkey. [2] Center for Computational Medicine and Bioinformatics, University of Michigan Medical School, Ann Arbor, MI, USA. [3] Department of Biomedical Sciences, School of Medicine and Health Sciences, University of North Dakota, Grand Forks, North Dakota, USA
[$]To whom correspondence should be made: junguk.hur@med.und.edu

Abstract

Drug labels contain detailed information about the drugs including their safety concerns which are regulated by the United States Food and Drug Administration (FDA). Adverse drug reactions (ADR) are adverse reactions associated with a specific drug. Automatic extraction of ADRs could help FDA greatly regulate drug safety. In this study, we employed an integrated approach of machine learning (ML)-based and dictionary-/rule-based methods to recognize ADR terms and normalize these terms to MedDRA Preferred Terms. The machine learning approach was used for the identification of the entities and is based on a recently proposed deep learning architecture. The model includes bi-directional Long Short-Term Memory (Bi-LSTM), a Convolutional Neural Network (CNN), and Conditional Random Fields (CRF). Alternatively, a dictionary- and rule-based approach was also used to identify ADR terms. MedDRA terms were added as a dictionary to SciMiner, our in-house text-mining system, and multiple rules for term expansion and exclusion to increase coverage and accuracy were implemented. The best performance was achieved using a combined approach: ADRs were first identified by the ML-based approach and then normalized to MedDRA Preferred Terms by the dictionary- and rule-based approach. Our system achieved 76.97% F1 score on the entity detection task and 82.58% micro-averaged F1 score on the ADR normalization task in the TAC 2017 ADR challenge.

Introduction

Pharmacovigilance is defined as “the science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other drug problem” (World Health Organization and others, 2002). It is impossible to know all possible adverse events of a particular drug, since generalizability of the clinical trials are low, sample sizes are small, and duration is short. FDA uses the Adverse Event Reporting System (FAERS) to detect adverse events. FEARS includes mandatory reports from pharmaceutical companies and reports that have been submitted to MedWatch directly. ADRs are still in the top 10 leading causes of death, and cost approximately $75 billion annually in the United States (Ahmad, 2003).

In addition to using medical reports (Gurulingappa, et al., 2011), it has been proposed to use data from social media (Leaman, et al., 2010), since users tend to discuss their sicknesses, treatments and also prescribed drugs and their effects in social media platforms. These discussions are not only confined to social networks specifically dedicated to health-related issues, but they also exist in generic platforms which could all be used for multi-corpus training to increase accuracy (Sarker & Gonzales, 2015). Preparing a lexicon (Leaman, et al., 2010) for detection of ADRs requires a lot of manual work and also limits a system's effectiveness to the extent of the lexicon. Nikfarjam and Gonzalez (Nikfarjam & Gonzalez, 2011) used syntactic and semantic patterns to remedy the shortcomings of lexicon-based approaches. Detailed information on ADR extraction with different techniques on various data sources is available in (Harpaz, et al., 2014) and (Karimi, et al., 2015).

The current approach for FEARS case report review requires manual reading of the text of the drug labels in order to determine whether a candidate ADR has been reported before or not. The automation of the extraction of the ADRs from drug labels would increase the efficiency of this process.

The TAC-ADR 2017 challenge targeted the automatic extraction of ADRs from drug labels and normalization of them through MedDRA (Medical Dictionary for Regulatory Activities) (Brown, et al., 1999), which is a dictionary for medical terminology. We participated in Task 1 (Extracting ADRs and related mentions from drug labels) and Task 4 (Linking the extracted ADRs to MedDRA terms). Mention can be defined as the portion of a text that corresponds to a certain entity such as an ADR. For example, given the sentence "Exclusive of an uncommon, mild injection site reaction, no adverse reactions to 11 C-choline have been reported." obtained from the drug label of choline, "injection site reaction" is an ADR mention and "mild" is a severity mention. While Task 1 addressed identifying these mentions, Task 4 targeted the normalization of the ADR mentions to MedDRA terms. The MedDRA preferred term for the ADR in the sentence above is "Injection site reaction" and its MedDRA preferred term ID is "10022095".

We investigated the integration of machine learning and dictionary/rule-based methods. Our best results were achieved by an integrated system that is based on a deep learning model for entity mention extraction and a dictionary/rule-based method for the normalization of the extracted ADRs to MedDRA terms. Our system and results are described in the following sections.

System Description

A high-level description of our integrated deep learning and dictionary/rule-based approach for entity detection and normalization is illustrated in Figure 1.

We investigated the performance of using both a machine learning approach and a dictionary/rule-based approach for Task 1 of the TAC-ADR 2017 challenge, whose goal was to extract entity mentions in drug labels such as *ADR*, *drug class*, *animal, severity*, *factor*, and *negation*. Mentions other than ADRs have only been annotated by human annotators, if they are related to any of the ADRs in the drug label. For example, in the sample sentence provided in the Introduction section, the severity mention “mild” has been annotated, since it defines the severity of the ADR “injection site reaction”. If “mild” occurs in a drug label in another context such as the symptoms of a disease being mild, then it is not annotated, since it is not related to an ADR.

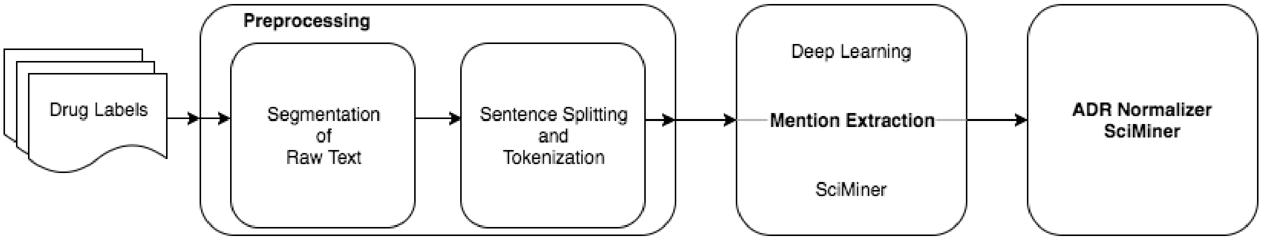


Figure 1. Overall workflow. Pre-processing only needed when Deep Learning architecture is used

We also participated in Task 4 of the challenge, which aimed to normalize the positive ADRs detected in Task 1 to their corresponding MedDRA terms. For ADR normalization we extended and used our in-house literature mining program SciMiner (Hur, et al., 2009), which is a dictionary- and rule-based literature mining platform for identification of genes and proteins in a context-specific corpus. MedDRA preferred terms (PT) and lowest level terms (LLT) were added to SciMiner, which normalized the positive ADRs to MedDRA preferred terms. MedDRA has the medical terminology hierarchy arranged from very specific to very general, where LLT is the most specific layer and PT is on top of it.

The machine learning component operates on sentence level and requires the input to be tokenized. Therefore, the first step of our system was to transform the drug labels, given in XML format, to sentence-split and tokenized format. The NLTK package (http://www.nltk.org) was used for sentence splitting and tokenization. Since the documents were not well formatted and contained tables, a Python script was internally prepared to detect text pieces and table parts. These initial preprocessing operations increased the performance of the sentence splitter.

The machine learning and dictionary-based components of the system are described in more detail in the following subsections.

Neural Network Architecture

A deep learning model designed for extracting NERs, which makes use of Bi-LSTM – CNN – CRF (Ma & Hovy, 2016), was used for the extraction of ADR mentions. We used the implementation proposed by (Reimers & Gurevych, 2017) which has minor differences from (Ma & Hovy, 2016). The model works on the sentence level, where every token is represented by a vector. Here, we describe the network starting from the creation of the input vectors to the prediction of the entity tags, which are calculated for every token of a given sentence.

Combined Word Embeddings

Every token in a given sentence was transformed into a vector before being fed into the model. These vectors consist of three parts, namely character embeddings, word embeddings, and case embeddings. The character embeddings were generated by a convolutional neural network (CNN) that runs over the characters of a given token. This representation has been shown to be powerful in encoding morphological information (Ma & Hovy, 2016), which we expect to be useful in the biochemical domain as well. At the first step, the tokens were transformed into their matrix representation by concatenating their character embeddings. Since CNNs work on fixed length input, all matrices were filled with padding to the length of the longest word in the vocabulary. Filter size was set to be 3 with a stride value of 1. In total 30 filters with these parameters were used for each input token in the CNN architecture. After using a max-pooling operation, a vector of length 30 was generated for each token. Figure 2 illustrates the workflow of the generation of character embeddings using the CNN component.

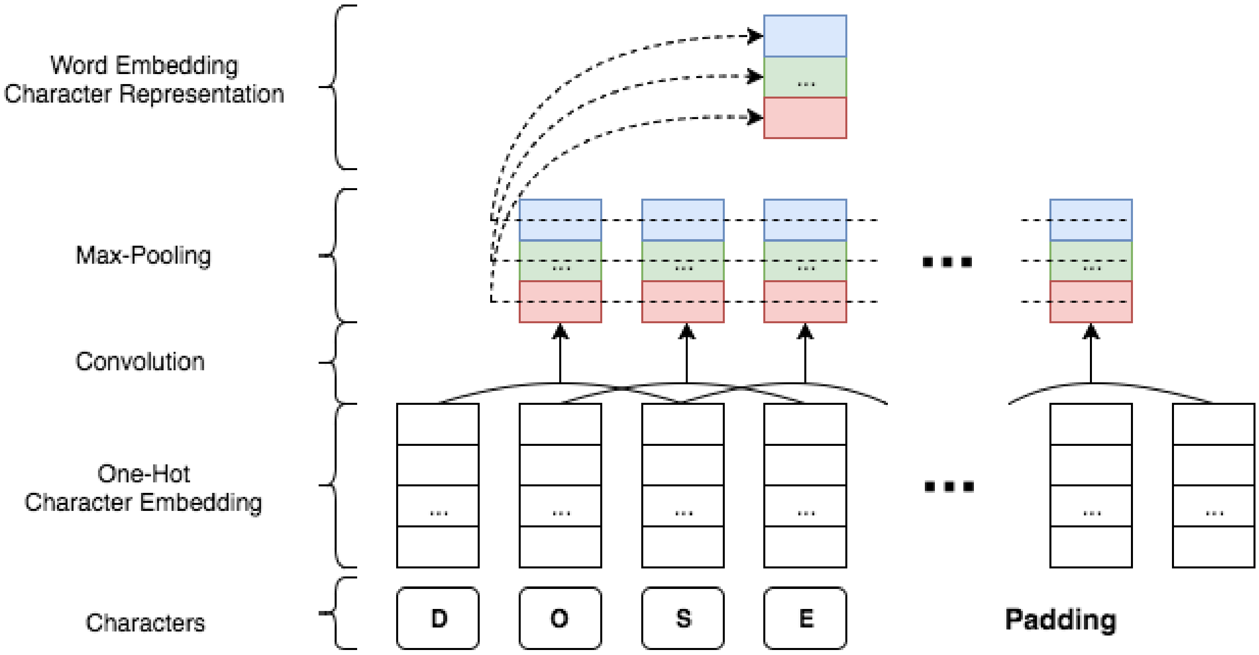
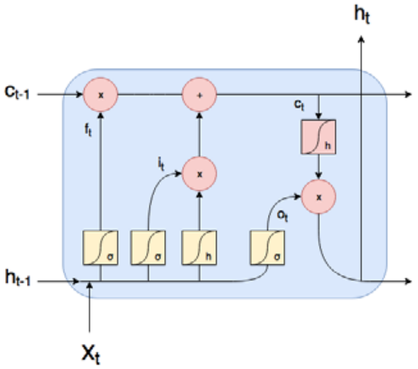


Figure 2. Character representation generation with CNN.

The word embeddings were generated by the Word2Vec tool (Mikolov, et al., 2013) in order to incorporate semantic information of words, since these representations had been shown to be effective in capturing semantic meanings (Mikolov, et al., 2013). The performance is expected to increase when these embeddings are generated from a corpus that is more related to the task; therefore, we used pre-trained embeddings that were generated using PubMed as the training corpus (Pyysalo, et al., 2013). These vectors of length 200 were appended to the character embeddings created by CNN. While looking for the vector representation of a token, our system also looked for lower cased and normalized versions in order to reduce out-of-vocabulary (OOV) words. However, it should be noted that this process decreased the number of OOV words, but we also lost the actual casing information of tokens. In order to remedy this loss, one-hot encoded case embeddings with length 8 were appended to the word embedding vectors, obtaining the combined word embedding vectors.

The Bi-LSTM and CRF Components



$$
\begin{aligned}
i_t &= \alpha(W_i[h_{t-1}, x_t] + b_i) \\
f_t &= \alpha\left(W_f[h_{t-1}, x_t] + b_f\right) \\
\tilde{c}_t &= tanh(W_c[h_{t-1}, x_t] + b_c) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c}_t \\
o_t &= \alpha(W_o[h_{t-1}, x_t] + b_o) \\
h_t &= o_t * \tanh(c_t)
\end{aligned}
$$

Figure 3. LSTM Component

Our model used a long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) component, which takes as input the combined word embeddings in order to model the context information for each word as shown in Figure 3. LSTM is from the family of Recurrent Neural Networks (RNNs), which are designed to learn patterns within sequences (Hochreiter & Schmidhuber, 1997). Even though these components are theoretically capable of learning long distance dependencies, it is hard to train them with gradient descent due to the problems of gradient vanishing or explosion (Bengio, et al., 1994). LSTMs are better in dealing with the gradient vanishing problem compared to the vanilla RNN, but they cannot solve the gradient explosion problem. As a solution to the gradient explosion problem, our model used gradient normalization (Pascanu, et al., 2013) with the value of 1, since it has been shown to be effective in the NER task (Reimers & Gurevych, 2017).

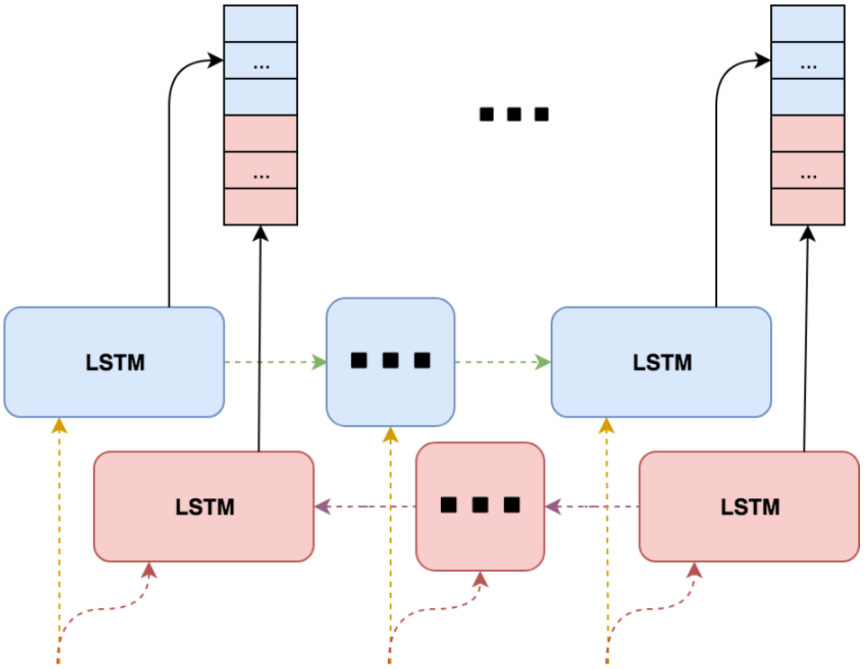


Figure 4. Bi-LSTM component with variational dropout (depicted by colored & dashed connections)

For detecting NERs, it has been shown to be an effective approach to have prior knowledge about the rest of the sentence well as the beginning. Two recent studies (Lample, et al., 2016; Ma & Hovy, 2016) used two LSTMs which run on opposite directions on the input sequences. Therefore, as shown in Figure 4, the outputs of the two LSTMs are concatenated. Two of these Bi-LSTM components are stacked. The first Bi-LSTM has 100 recurrent units and the second one has 75 recurrent units.

Dropout (Srivastava, et al., 2014) is a way to prevent overfitting in neural networks. However it has been shown to be difficult to apply on RNN layers (Gal & Ghahramani, 2015). Hence, variational dropout (Gal & Ghahramani, 2015) has been applied in the Bi-LSTM layers. This method applies the same mask through time in recurrence, which is shown by colored dashed arrows in Figure 4. Dropout of 0.25 was applied in our Bi-LSTM components.

The last layer is the Conditional Random Fields (CRF) (Lafferty, et al., 2001), which does the prediction of the token tags. The TAC-ADR dataset contained non-contiguous mentions such as “Interstitial infiltration ... of the chest” with 10 words, but CRF is expected to work better if all mentions are contiguous.

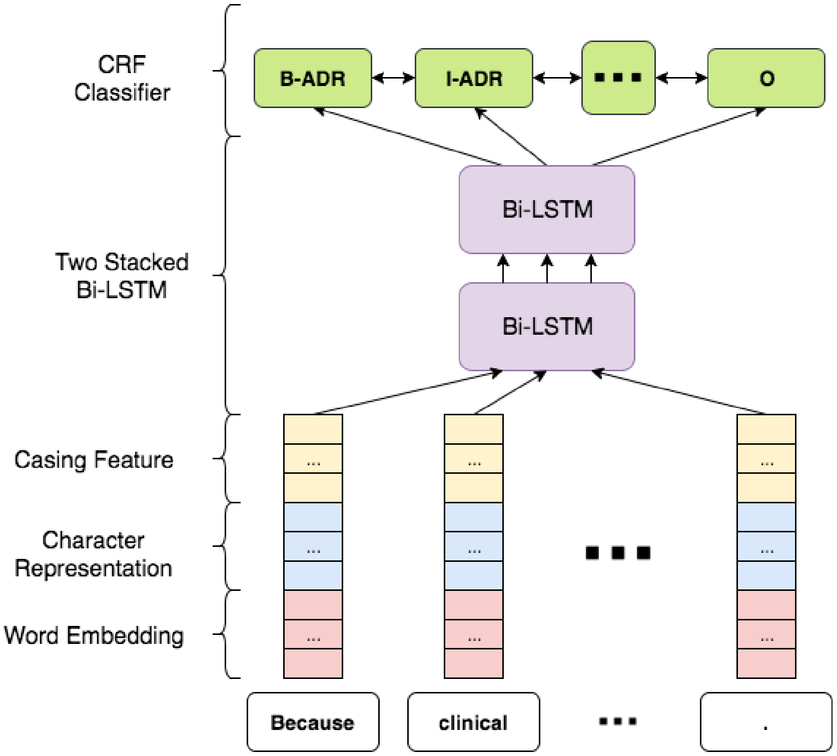


Figure 5. Deep learning model for NER

The CNN Bi-LSTM and CRF models are combined and used as the final deep learning model as shown in Figure 5. The NADAM (Dozat, 2016) optimization technique is used in the training of the combined model.

SciMiner

In parallel to the neural network-based approach above, we employed a dictionary- and rule-based Named Entity Recognition (NER) identification approach. We used SciMiner written in Perl, which was originally developed as a web-based literature mining platform for identifying genes and proteins in biomedical literature (Hur, et al., 2009). SciMiner has been expanded to identify various biomedical ontologies such as Vaccine Ontology (VO) and Interaction Network Ontology (INO), developed by our group, resulting in specific variations of SciMiner: INO-SciMiner (Hur, et al., 2015), VO-SciMiner (Hur, et al., 2011), and E-coli-SciMiner (Hur, et al., 2017).

In this study, we added the MedDRA preferred and lowest level terms (PT and LLT, respectively) to SciMiner. Manual review of these terms was also performed to identify such terms that are unlikely to be ADRs such as various cancers. Various rules for term expansion as well as exclusion to increase coverage and accuracy were implemented. For example, Perl library Lingua::EN was used to expand the base ADR dictionary allowing the inclusion of additional plural or singular forms, when only one form was included in the base dictionary. SciMiner-based approach was also used for normalizing the positive ADR terms, identified by the deep learning-based approach in the above section, to their respective MedDRA PTs.

Experiments and Results

Dataset

The training set consisted of XML formatted 101 drug label files. These XML files contained raw texts with sections, mentions, relations and normalizations for reactions. Our team, named as “CONDL”, participated in the first and fourth tasks of the TAC-ADR 2017, which have the aims to extract the mentions from a given drug label (Task1) and normalize them to appropriate MedDRA PTs (Task 4). SciMiner worked on the raw text directly, whereas the deep learning model worked at the sentence level; therefore, the text had to be split first as the initial process. We used NLTK (Bird, et al., 2009) sentence splitter and tokenizer.

We used the NLTK tokenizer (Bird, et al., 2009) to identify the tokens in the sentences and transformed every drug label file into the CoNLL format. The sentences were separated by an empty line and every token was written on a separate line. An example sentence is shown in Table 1 and its CoNLL format is shown in Table 2, where every line consists of 6 columns and starts with the token itself. The second column holds the tag type of the token, which was encoded with BIO2 (Sang & Veenstra, 1999) chunking representation. “B” denotes that the token is the beginning of an entity mention, “I” denotes that the token is inside of a mention, and "O" (Outside) indicates that the token is not part of a mention. For example, the tags of an ADR term "hypersensitivity reactions" are "B-ADR I-ADR" according to this representation. The third column holds the Part-Of-Speech (POS) values of each token, which was not utilized by the current model. The following columns show the location of the token within a label. The first one of those is the id of the section. The second one is the start position of the token within the section and the last one shows the length of the token.

Normalization was done by SciMiner, which works on the strings of the detected ADR mentions.

Raw Text	Long-term cumulative radiation exposure is associated with an increased risk for cancer.
Related Mentions	<**Mention id**="M10" **section**="S2" **type**="Factor" **start**="2309" **len**="4" **str**="risk" /> <**Mention id**="M11" **section**="S2" **type**="AdverseReaction" **start**="2318" **len**="6" **str**="cancer" />
Related Reaction	<**Reaction id**="R4" **str**="cancer"> <**Normalization** id="R4.N1" **meddra_pt**="Neoplasm malignant" **meddra_pt_id**="10028997" **meddra_llt**="Cancer" **meddra_llt_id**="10007050" /> </**Reaction**>

Table 1. Example sentence for drug label "choline"

	Raw Text	BIO encoding	POS tag	Section	Offset	Length
CoNLL format (BIO encoding)	Long-term	**O**	JJ	S2	2237	9
	cumulative	**O**	JJ	S2	2247	10
	radiation	**O**	NN	S2	2258	9
	exposure	**O**	NN	S2	2268	8
	is	**O**	VBZ	S2	2277	2
	associated	**O**	VBN	S2	2280	10
	with	**O**	IN	S2	2291	4
	an	**O**	DT	S2	2296	2
	increased	**O**	VBN	S2	2299	9
	risk	**B-FAC**	NN	S2	2309	4
	for	**O**	IN	S2	2314	3
	cancer	**B-ADR**	NN	S2	2318	6
	.	**O**	.	S2	2324	1

Table 2 Transformation of sentence in Table 1

Results

For the workshop evaluation, we submitted three sets of results: CONDL1, CONDL2, and CONDL3. Table 3 summarizes the approaches taken in each set and Table 4 shows the obtained results.

Set	Named Entity Recognition	ADR Normalization
CONDL1	ML	SciMiner
CONDL2	SciMiner	SciMiner
CONDL3	SciMiner + non-ADRs from ML	SciMiner

Table 3. Summary of approaches

These three sets accomplished overall F1-measures ranging from 67.4% to 77.0% in NER identification (Task 1), and micro-level F1-measures between 77.6% to 82.6% and macro-level F1-measures between 75.6% and 80.5%) in normalizing to appropriate MedDRA PT, respectively (Task 4). The best performance was achieved when NERs were identified using our ML approach, which were then normalized to MedDRA Preferred Terms by dictionary- and rule-based approach (SciMiner).

			CONDL1	**CONDL2**	**CONDL3**
Task 1	**+type**	**Precision**	76.5	65.5	65.2
		Recall	77.5	61.4	69.8
		F1	77.0	63.4	67.4
	-type	**Precision**	76.5	65.5	65.2
		Recall	77.5	61.4	69.8
		F1	77.0	63.4	67.4
Task 4	**micro**	**Precision**	88.8	74.6	74.6
		Recall	77.2	81.0	81.0
		F1	82.6	77.6	77.6
	macro	**Precision**	88.2	73.1	73.1
		Recall	75.8	79.9	79.9
		F1	80.5	75.6	75.6

Table 4 Official evaluation results

Conclusion

In this paper, we employed two different methods for detecting mentions of type ADR, drug class, animal, severity, factor, and negations from drug labels. The neural network-based approach outperformed the dictionary- and rule-based approach in extracting ADRs.

As future work we will investigate incorporating ontology and dictionary knowledge into the deep learning model. Also updating the word embeddings (Chiu, 2016), making an extensive parameter search and solving the problems with chunk labeling and preprocessing are likely to increase the performance of the deep learning model.

Acknowledgement

We would like to thank Mr. Mehmet Köse at the Bogazici University for his help in data preprocessing at the initial phase of the project.

Bibliography

Ahmad, S. R., 2003. Adverse drug event monitoring at the Food and Drug Administration. *Journal of general internal medicine,* Volume 18, pp. 57-60.

Bengio, Y., Simard, P. & Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE,* Volume 5, pp. 157-166.

Bird, S., Loper, E. & Klein, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit.

Chiu, B. a. C. G. a. K. A. a. P. S., 2016. How to train good word embeddings for biomedical NLP. *Proceedings of BioNLP16,* p. 166.

Dozat, T., 2016. Incorporating nesterov momentum into adam.

Gal, Y. & Ghahramani, Z., 2015. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *arXiv preprint,* pp. 1019-1027.

Gal, Y. & Ghahramani, Z., 2015. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *arXiv: Machine Learning,* pp. 1019-1027.

Gurulingappa, H., Fluck, J., Hofmann-Apitius, M. & Toldo, L., 2011. *Identification of adverse drug event assertive sentences in medical case reports.* s.l., s.n., pp. 16-27.

Harpaz, R. et al., 2014. Text Mining for Adverse Drug Events: the Promise, Challenges, and State of the Art. 1 10.10(37).

Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation,* 9(8), p. 1735–1780.

Hur, J., Özgür, A. & He, Y., 2017. Ontology-based literature mining of E. coli vaccine-associated gene interaction networks. *Journal of biomedical semantics,* 8(1), p. 12.

Hur, J., Özgür, A., Xiang, Z. & He, Y., 2015. Development and application of an interaction network ontology for literature mining of vaccine-associated gene-gene interactions. *Journal of biomedical semantics,* 6(1), p. 2.

Hur, J., Schuyler, A. D., States, D. J. & Feldman, E. L., 2009. SciMiner: Web-based literature mining tool for target identification and functional enrichment analysis. *Bioinformatics,* 25(6), pp. 838-840.

Hur, J., Xiang, Z., Feldman, E. L. & He, Y., 2011. Ontology-based Brucella vaccine literature indexing and systematic analysis of gene-vaccine association network. *BMC immunology,* 12(1), p. 49.

Karimi, S. et al., 2015. Text and Data Mining Techniques in Adverse Drug Reaction Detection. May, 47(4), p. 56:1–56:39.

Lafferty, J., McCallum, A. & Pereira, F. C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lample, G. et al., 2016. Neural architectures for named entity recognition. *arXiv preprint.*

Leaman, R. et al., 2010. *Towards Internet-age Pharmacovigilance: Extracting Adverse Drug Reactions from User Posts to Health-related Social Networks.* Stroudsburg, Association for Computational Linguistics, p. 117–125.

Ma, X. & Hovy, E. H., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *arXiv preprint,* Volume 1, pp. 1064-1074.

Mikolov, T. et al., 2013. Distributed Representations of Words and Phrases and their Compositionality. *arXiv preprint,* pp. 3111-3119.

Nikfarjam, A. & Gonzalez, G. H., 2011. Pattern Mining for Extraction of mentions of Adverse Drug Reactions from User Comments. *AMIA Annual Symposium Proceedings,* Volume 2011, pp. 1019-1026.

Pascanu, R., Mikolov, T. & Bengio, Y., 2013. *On the difficulty of training recurrent neural networks.* s.l.:International Conference on Machine Learning.

Pyysalo, S. et al., 2013. *Distributional Semantics Resources for Biomedical Text Processing.* [Online]
Available at: http://escholar.manchester.ac.uk/uk-ac-man-scw:267174
[Accessed 23 10 2017].

Reimers, N. & Gurevych, I., 2017. *Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging.* s.l.:arXiv preprint.

Sang, E. F. & Veenstra, J., 1999. *Representing text chunks.* s.l., Association for Computational Linguistics, pp. 173-179.

Sarker, A. & Gonzales, G., 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of Biomedical Informatics,* 1 February, Volume 53, pp. 196-207.

Srivastava, N. et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from .... *Journal of Machine Learning Research,* 15(1).

Tomas, M., 2012. *Statistical language models based on neural networks.* s.l.:Brno University of Technology.

World Health Organization and others, 2002. *The importance of pharmacovigilance,* s.l.: Geneva: World Health Organization.