Proceedings of Machine Learning Research 90:7–15, 2018 Medication and Adverse Drug Event Detection Workshop

Named Entity Recognition using Neural Networks for Clinical Notes

Edson Florez* florez@i3s.unice.fr
Frederic Precioso precioso@unice.fr
Michel Riveill riveill@unice.fr
Université Cote d'Azur, CNRS, I3S, Nice 06000, France

Romaric Pighetti romaric.pighetti@francelabs.com
France Labs, Saint-Laurent-du-Var 06700, France

Editor: Feifan Liu, Abhyuday Jagannatha, Hong Yu

Abstract

Currently, the best performance for Named Entity Recognition in medical notes is obtained by systems based on neural networks. These supervised systems require precise features in order to learn well fitted models from training data, for the purpose of recognizing medical entities like medication and Adverse Drug Events (ADE). Because it is an important issue before training the neural network, we focus our work on building comprehensive word representations (the input of the neural network), using character-based word representations and word representations. The proposed representation improves the performance of the baseline LSTM. However, it does not reach the performances of the top performing contenders in the challenge for detecting medical entities from clinical notes (Yu et al., 2018).

Keywords: Named Entity Recognition, Clinical Notes, Adverse Drug Events, Deep Learning, LSTM.

1. Introduction

Patients are often subject to multiple treatments, which may be the cause of adverse effects. Therefore, it is necessary to establish if an Adverse Event (AE) has occurred after taking medicines. AE refers to any adverse event occurring at the time a drug is used, whether it is identified as a cause of the event or not. In case one can establish a relation between the AE and the drug, then the relation is considered as an Adverse Drug Event (ADE) or Adverse Drug Reaction (ADR).

For the purpose of identifying ADE mentions, we use medical notes provided in EHR (Electronic Health Records). These notes contain mentions of medical entities like medications, ADE (Adverse Drug Event) and symptoms. These terms have to be identified and classified in the right category. This classification problem is known as Named Entity Recognition (NER). It has been performed with Machine Learning and Deep Learning algorithms to classify entities into categories such as ADE and medications in medical notes. In this work neural networks are used for NER in clinical notes, using several word representation together to improve the performance. Section 2 presents some related works in

* The Provence-Alpes-Cote d'Azur region and the France Lab company finance the doctoral work.

the NER domain. The several features used as well as the network used are explained in Section 3. Finally, in Section 4 the models performance using the dataset provided by the MADE1.0 Challenge (Yu et al., 2018) is presented.

2. Related Work

Conditional Random Fields (CRFs) is a machine learning algorithm used for ADR extraction (Liu et al., 2017), with context features around the current word (Nikfarjam et al., 2015). It takes every neighbour word in a fixed window of words. Other Machine Learning algorithms like Support Vector Machines (SVMs) are commonly used for NER. Gurulingappa et al. (2012) built a system for the identification and extraction of potential adverse events of drugs with SVM. Their dataset is an ADE corpus from MEDLINE (Medical Literature Analysis and Retrieval System Online) case reports that are manually annotated. The corpus contains annotations for the mentions of drugs, ADE, and relations between drugs and medical conditions representing clear adverse reactions (relation drug-cause-condition).

The CLEF Challenge provides system performance for NER using the QUAERO French Medical Corpus (Névéol et al., 2014). It has ten categories for annotations of medical entities, with data collected from the EMEA (European Medicines Agency) documents and titles of research articles indexed in the MEDLINE database. A Dictionary-based concept recognition system overcame CRF and SVM classifiers in CLEF 2015 (Névéol et al., 2015) on the MEDLINE corpus, according to the Exact Match metric, which considers a term (word or group of words that have a label) as correctly classified only if all the words in the term received the correct label.

Deep learning models like CNN (Convolutional Neural Network) are used to detect the presence of ADR (Huynh et al., 2016), such as in binary classification problem on two datasets (from Twitter and case reports (Gurulingappa et al., 2012)). Overall, CNN appears to perform better compared to other more complex CNN variants that have a RNN (Recurrent Neural Network) layer (Gurulingappa et al., 2012). However, CCNA (Convolutional Neural Network with Attention) is better on the dataset of case reports. Overall, results on the case reports are better than those on the Twitter dataset. Tweets contain a lot of ill-grammatical sentences and short forms (Huynh et al., 2016) that hinders the performances, which highlights the importance of de-noising the data.

The adverse event detection problem focused on clinical notes is a sequential problem, and RNN models are specialized for it because at time step t, the recurrent node takes as input the outputs produced by the previous state. Simple RNN models can classify the input sequence taking into account the long time dependencies (Liwicki et al., 2007), but they face the problem of vanishing gradients (Bengio et al., 1994), instead another RNN architecture known as Long Short-Term Memory (LSTM), reduces the impact of this problem using a short memory connection along the sequence. LSTM was applied to sequential problems such as Handwriting Recognition (Liwicki et al., 2007) and Named Entity Recognition (Jagannatha and Yu, 2016a). LSTM exploits the long term label dependencies for sequence labelling in clinical text, e.g. in the sentence "the patient has internal bleeding (ADE) secondary to warfarin (Medication)", the label for ADE is strongly related to the label prediction of Medication, then Warfarin is labelled as Medication using information of previous ADE tag (internal bleeding), which is stored in the memory of LSTM cells.

LSTM was used with an annotated corpus of English Electronic Health Records (EHR) from cancer patients in Jagannatha and Yu (2016b), with labels for several medical entities (like Adverse Drug Event (ADE), drug name, dosage) and relations between entities. The best LSTM version in Jagannatha and Yu (2016b) is the Approximate Skip Chain CRF-RNN network, which implements a CRF algorithm after the bidirectional LSTM output. This network has a high precision for DrugName detection, but a low precision for ADE, probably because the dataset in unbalanced and has less ADE samples.

Results of NER algorithms dedicated to ADE detection are collected in the review article (Sarker et al., 2015). This review shows that Machine learning and Deep Learning algorithms are outstanding at this task. However, the performance presented in this review were obtained on different datasets, making the comparison somewhat unfair. Comparing the best result reported in Huynh et al. (2016) and Gurulingappa et al. (2012) using the same dataset (last lines of Table 1), one can observe that Gurulingappa et al. (2012) obtained slightly better results on Recall, Precision and F-score.

Table 1: Methods for ADE extraction.

Study	**Method**	**Size**	**Rec.**	**Prec.**	**F1**
Nikfarjam and Gonzalez (2011)	Lexical pattern-matching	1200	0.66	0.70	0.68
Nikfarjam et al. (2015)	Supervised learning via Conditional Random Fields (CRFs)	1559	0.78	0.86	0.82
Jagannatha and Yu (2016b)	Bi-LSTM-CRF (Skip-CRF-Approx.)	1154	0.83	0.81	0.82
Huynh et al. (2016)*	CNNA (Convolutional Neural Network with Attention)	2972	0.84	0.82	0.83
Gurulingappa et al. (2012)*	SVM (Support Vector Machines)	2972	**0.86**	**0.89**	**0.87**

Note: * Systems using the same dataset

LSTM model has shown to be appropriate on the state of the art for sequential problems. However, in order to improve performance, it is important to feed the network with an appropriate input representation (an embedding) (Chiu and Nichols, 2015). This representation replaces each unique word with a dense vector representation, which tries to provide closer vectors among word synonyms or related words. In Jagannatha and Yu (2016b) the embedding layer values used were initialized using a skip-gram word embedding. The skip-gram embedding was calculated using unlabelled data from PubMed open access articles, English Wikipedia and an unlabelled EHR corpus. We can also improve the precision of LSTM with additional features for its input, such as character-level features from each word extracted using CNN or LSTM (Liu et al., 2017), and then concatenate character and word representations inspired by the work of Chiu and Nichols (2015). All this was implemented in our work, as described in the following section.

3. Model

In our final model, we use a comprehensive word representation, which concatenates character-level representations, word embedding and POS features. This is described in the following subsections, as well as the full network using that representation to solve the NER task.

3.1. Features

The character-level features can exploit prefix and suffix information about words (Lampele et al., 2016), to have closer representations among words of the same category. This is particularly useful for terms that may be Out-Of-Vocabulary (OOV), i.e. words that appear in the test data and not in the training data. OOV is a common issue with domain specific words, and prefix and suffix representations can help a lot. For example, the words "Clonazepam" and "Lorazepam" both belong to the medication category in the medical context and may be OOV. However they share the same suffix, making them closer to each other on a character-level feature. Therefore we build a LSTM network (see subsection 2.2) that get representations of words based on their characters.

Another feature used is Part-of-speech (POS), which tags the words with labels like noun, verb, adjective, adverb, etc. It classifies words according to its roles within the grammatical structure of the sentence. Medications for example will always belong to the Noun category, making them close together with respect to this feature. The tagging was performed using an Averaged Perceptron algorithm [1].

Finally, we also use word embeddings learned from a large corpus, to consider the contexts in which words appear usually. It can create similar vectors (representations) for words that appear in similar contexts, such as the names of different countries. The word embedding of dimension 200 provided by Jagannatha and Yu (2016b) were used, as well as another of 300 dimensions provided by FastText (Bojanowski et al., 2016). Both are pretrained with skip-gram using unlabeled data mainly from Wikipedia.

3.2. Network Description

Long Short-term Memory Networks (LSTMs) can learn long term dependencies among the words in the sentence (Jagannatha and Yu, 2016b). LSTM keeps information in a memory-cell that is updated using input and forget gates (Lample et al., 2016).

The character-level embedding for words was built by a Bi-LSTM network (represented on the bottom left of Figure 1). First, each character takes an integer value from a lookup table, then it is replaced by a one-hot vector. The final state of the forward and backward LSTM is the representation of the suffix and prefix of the word. The Character-level embedding is the concatenation of both LSTM layers, so with LSTM layers of 20 cells (units), we get a vector of 40 dimensions. This character-level representation is concatenated to the word embedding and the POS feature to form the final comprehensive word representation (see Figure 1) (Lample et al., 2016).

The comprehensive word embedding is the input of a Bi-LSTM network, which takes a sequence of words and returns a sequence of hidden states at every time step (see Figure 2). The raw sentence is processed with a regular expression tokenizer into sequence of tokens. Sentences longer than the sequence length were cropped to size, and shorter sentences were pre-padded with masks. The forward and backward LSTM layers get hidden state sequences, which represent the left and right context of the sentence at every time step (word), and their concatenation is the representation of a word in context (Graves and Schmidhuber, 2005).

The bidirectional LSTM provides scores for every possible label for each word, its output

1. See https://www.nltk.org/api/nltk.tag.html

(hidden states) feed the inference layer for tagging each word independently (see Figure 2). For that, the hidden states are connected by a dense layer (i.e. fully connected layer) to each possible label, and a Softmax function over the score of all possible labels produces a probability for each label (values between 0 and 1 that together sum 1), which is used to get the predicted label. The predictions (labels probabilities) of the Softmax output is evaluated with the correct class (true labels). The target labels consist in an integer vector where each element represents the position of the number 1 in a one-hot encoding. Categorical Cross-entropy is the loss function used, which penalizes the deviation between the predicted and target (true) labels during training. Then the optimization function will minimize the loss of the correct labels sequence.

For training, the input and output of the network will be the sequence of words (each word replaced by its comprehensive word representation) and its corresponding labels (see Figure 2), and LSTM will try to learn a model that minimize the error of the predicted label.

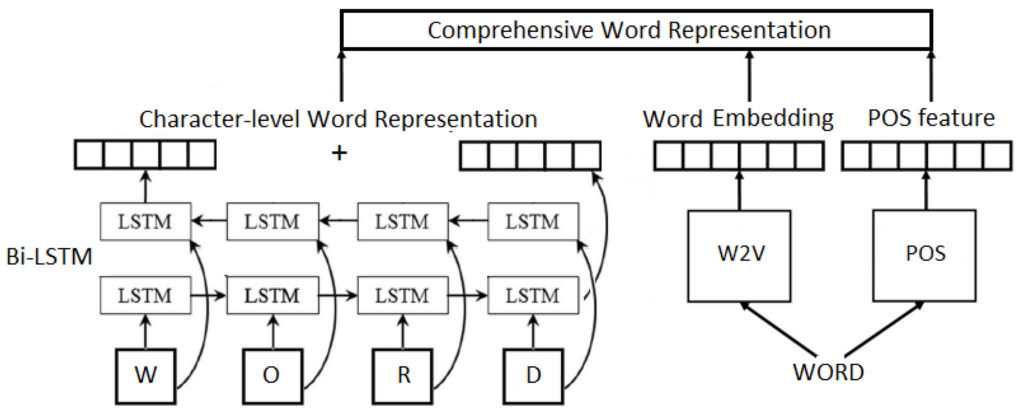

Figure 1: Comprehensive word representation

4. Results and Discussion

The dataset for our experiments was provided by the MADE1.0 Challenge (Yu et al., 2018). It was created with 1092 EHR notes from 21 cancer patients (Jagannatha and Yu, 2016b). It contains annotations of ADEs, indications, other signs and symptoms, medication, dosage, route, frequency, duration, severity. These annotations are used in the Named entity recognition (NER) task, and the dataset also has relations among those medical entities for the Relation Extraction task, like the relation Adverse between Medication and ADE annotations. In the NER task, the goal is identify and annotate the medical entities found in the raw clinical notes.

The models were compared with the same parameters and training dataset as those of the MADE challenge. The dataset is split into training (80% of the data) and testing (20% of the data). The results are shown in Table 2, with results for models with random initialization of word vectors (baseline), W2V of 200 dim (Jagannatha and Yu, 2016b), W2V(FT)

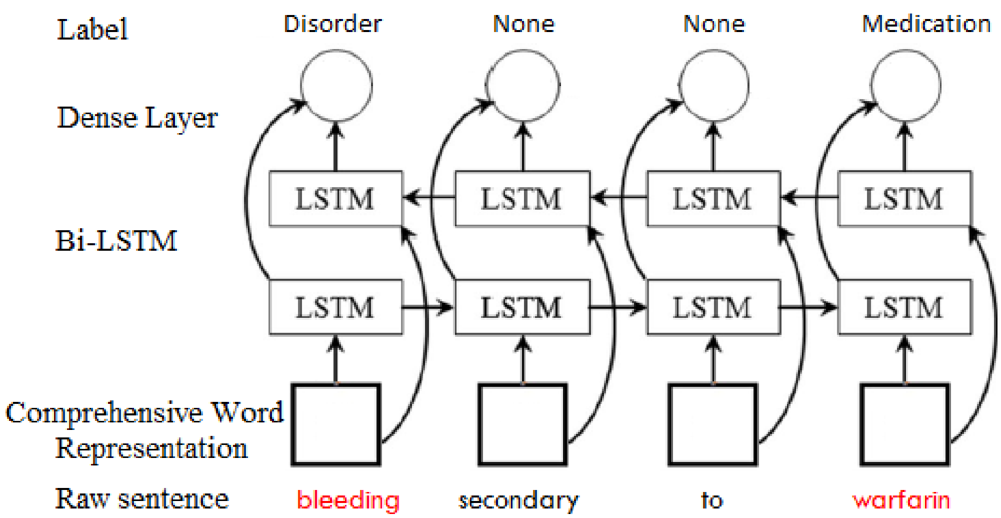

Figure 2: LSTM network for tagging

FastText of 300 dim (Bojanowski et al., 2016), POS features (46 tags) and Character-level word representation Char(LSTM) of length 40.

We improved the performance using the word embedding of FastText (W2V(FT)) more than using the one of W2V (Jagannatha and Yu, 2016b): FastText (W2V(FT)) got about 0.22 more in F1 than W2V (Jagannatha and Yu, 2016b). We observed the highest improvement over the baseline (randomly initialized model) with character-level representations and POS tags together, it increases the F1 of about 0.2. W2V(FT) only with the Char(LSTM) provides a small increase in F1, while POS alone does not increase anything.

Table 2: Performances of models for NER.

Model	**Recall**	**Prec.**	**F1**
Baseline	0,686	0,704	0,695
W2V(Jagannatha and Yu, 2016b)	0,668	0,689	0,678
Char(LSTM) + POS	0,659	0,678	0,668
W2V(FT)	0,694	0,721	0,707
W2V(FT) + POS	0,691	0,719	0,704
W2V(FT) + Char(LSTM)	0,692	**0,724**	0,708
W2V(FT) + Char(LSTM) + POS	**0,700**	0,721	**0,710**

Note: Parameters batch size 32, sequence length 60, 100 LSTM cells, learning rate 0.1

The best model (W2V+Char(LSTM)+POS) was trained with 100% of the training files, then it created the predicted annotations for the test dataset of the MADE Challenge. Table 3 shows the official results validated by the MADE challenge, the best result of 2 runs for standard (W2V (Jagannatha and Yu, 2016b)) and extended evaluation (W2V(FT)).

The usage of more hidden units (200 or 300 LSTM cells) did not significantly influenced the model performance, and big values (60, 70, 80) of the sequence length (number of words by sequence) gave better results in our experiments with the clinical notes of MADE dataset. The most appropriate initial value for the learning rate was 0.1, a smaller learning rate decreased the performance and increased the running time. The results are good but an additional strategy is still necessary to reach top performance systems (the best has 0.829 in F1). An additional layer of conditional random fields used over the output of LSTM (in the tagging layer), which takes into account the dependencies between labels to get an accurate score like in Jagannatha and Yu (2016b) would be interesting to test.

Table 3: Performances of models for NER task in MADE Challenge.

Model	**Recall**	**Prec.**	**F1**
W2V[1] + Char(LSTM) + POS	0,720	0,681	0,700
W2V(FT) + Char(LSTM) + POS	**0,748**	**0,716**	**0,732**

5. Conclusions

We implemented a LSTM network to solve the named entity recognition problem found on the Adverse Drug Reaction detection. This neural network requires good input features for training, so we built character-level features extracted with another LSTM, that were used in conjunction with word representations as a comprehensive word representation. This conjunction of features increased the performance of the LSTM, but it does not allow the LSTM alone to reach the best performance achieved for the task. Therefore, as future work, investigating the use of an additional technique for the network, as the Attentional model for RNN that gives more weight to words that are more important, sounds promising.

Acknowledgments

This work is partly funded by the French government labelled PIA program under its IDEX UCAJEDI project (ANR-15-IDEX-0001).

References

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. 5(2):157–166, 1994.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

J. Chiu and E. Nichols. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*, 2015.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN'05*, volume 4, pages 2047–2052. IEEE, 2005.

H. Gurulingappa, A. Mateen-Rajpu, and L. Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3(1):15, 2012.

T. Huynh, Y. He, A. Willis, and S. Rüger. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887, 2016.

A. Jagannatha and H. Yu. Bidirectional rnn for medical event detection in electronic health records. In *Proceedings of NAACL-HLT'16*, volume 2016, page 473. NIH Public Access, 2016a.

A. Jagannatha and H. Yu. Structured prediction models for rnn based sequence labeling in clinical text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 856. NIH Public Access, 2016b.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Learning long-term dependencies with gradient descent is difficult. *arXiv preprint arXiv:1603.01360*, 2016.

Z. Liu, M. Yang, X. Wang, Q. Chen, B. Tang, Z. Wang, and H. Xu. Entity recognition from clinical texts via recurrent neural network. *BMC medical informatics and decision making*, 17(2):67, 2017.

M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.

A. Névéol, C. Grouin, J. Leixa, S. Rosset, and P. Zweigenbaum. The quaero french medical corpus: A ressource for medical entity recognition and normalization. In *Proceedings of Fourth Workshop on Building and Evaluating Ressources for Health and Biomedical Text Processing - BioTxtM2014*, pages 24–30. Citeseer, 2014.

A. Névéol, C. Grouin, X. Tannier, T. Hamon, L. Kelly, L. Goeuriot, and P. Zweigenbaum. Clef ehealth evaluation lab 2015 task 1b: Clinical named entity recognition. In *CLEF (Working Notes)*, 2015.

A. Nikfarjam and G. Gonzalez. Pattern mining for extraction of mentions of adverse drug reactions from user comments. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1019. American Medical Informatics Association, 2011.

A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, and G. Gonzalez. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681, 2015.

A. Sarker, R. Ginn, A. Nikfarjam, K. O'Connor, K. Smith, S. Jayaraman, T. Upadhaya, and G. Gonzalez. Utilizing social media data for pharmacovigilance: a review. *Journal of biomedical informatics*, 54:202–212, 2015.

H. Yu, A. Jagannatha, F. Liu, and W. Liu. Nlp challenges for detecting medication and adverse drug events from electronic health records. *https://bio-nlp.org/index.php/announcements/39-nlp-challenges*, 2018.