Detecting Tweets Mentioning Drug Name and Adverse Drug Reaction with Hierarchical Tweet Representation and Multi-Head Self-Attention

Chuhan Wu[1], **Fangzhao Wu**[2], **Junxin Liu**[1], **Sixing Wu**[1], **Yongfeng Huang**[1] **and Xing Xie**[2]
[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Microsoft Research Asia
{wuch15,ljx16,wu-sx15,yfhuang}@mails.tsinghua.edu.cn
fangzwu,xing.xie@microsoft.com

Abstract

This paper describes our system for the first and third shared tasks of the third Social Media Mining for Health Applications (SMM4H) workshop, which aims to detect the tweets mentioning drug names and adverse drug reactions. In our system we propose a neural approach with hierarchical tweet representation and multi-head self-attention (HTR-MSA) for both tasks. Our system achieved the first place in both the first and third shared tasks of SMM4H with an F-score of 91.83% and 52.20% respectively.

1 Introduction

Social media services such as Twitter have become important platforms for information sharing and dissemination. Automatically detecting tweets which mentions drug names (DNs) and adverse drug reactions (ADRs) at a large scale is an interesting research topic and has many important applications such as pharmacovigilance (Sarker and Gonzalez, 2015; Han et al., 2017; Weissenbacher et al., 2018). However, tweets are very noisy and informal, and full of misspellings (e.g., "aspirn" for "aspirin") and user-created abbreviations (e.g., "COC" for "Cocaine"). In addition, many DN and ADR mentions are context-dependent. For example, "I take Vitamin C after meals" is a tweet mentioning drug name, but the tweet "Vitamin C is good for health" is not. Thus, the detection of DN and ADR mentioning tweets is very challenging.

In order to facilitate the research on automatic detection of tweets mentioning DN and ADR, two related shared tasks were released by the third Social Media Mining for Health Applications (SMM4H) workshop[1] (Weissenbacher et al., 2018). Task 1 aims to classify whether a tweet mentions any drug names or dietary supplement. Task 3 aims to classify whether a tweet contains adverse drug reaction mention. We designed a neural approach with hierarchical tweet representation and multi-head self-attention (HTR-MSA) to participate in these two tasks. Our hierarchical tweet representation model first learns word representations from characters using convolutional neural network (CNN) and then learns tweet representations from words using a combination of Bi-directional long-short term memory (Bi-LSTM) network and CNN. In addition, we incorporated additional features to enhance the word representations, including pre-trained word embedding, part-of-speech (POS) tag embedding, sentiment features based on sentiment lexicons and lexicon features extracted from medical lexicons. Besides, we applied multi-head self-attention mechanism to our approach to enhance the contextual representations of words by capturing the interactions between all words in tweets. Our system achieved 91.83% F-score in Task 1 and 52.20% F-score in Task 3, and ranked 1st in both task. The codes of our system are publicly available[2].

2 Our Approach

The architecture of our HTR-MSA model is shown in Fig. 1. It contains three major modules, i.e., word representation, tweet representation and tweet classification.

2.1 Word Representation

In order to handle the massive misspellings and user-created abbreviations of drug names in tweets, we propose to learn word representations from characters. There are three sub-modules in the word representation module.

The first one is character embedding, which converts each word from a sequence of characters into a sequence of low-dimensional dense vectors

[1]https://healthlanguageprocessing.org/smm4h/

[2]https://github.com/wuch15/SMM4H_THU_NGN

Proceedings of the 3rd Social Media Mining for Health Applications (SMM4H) Workshop & Shared Task, pages 34–37
Brussels, Belgium, October 31, 2018.

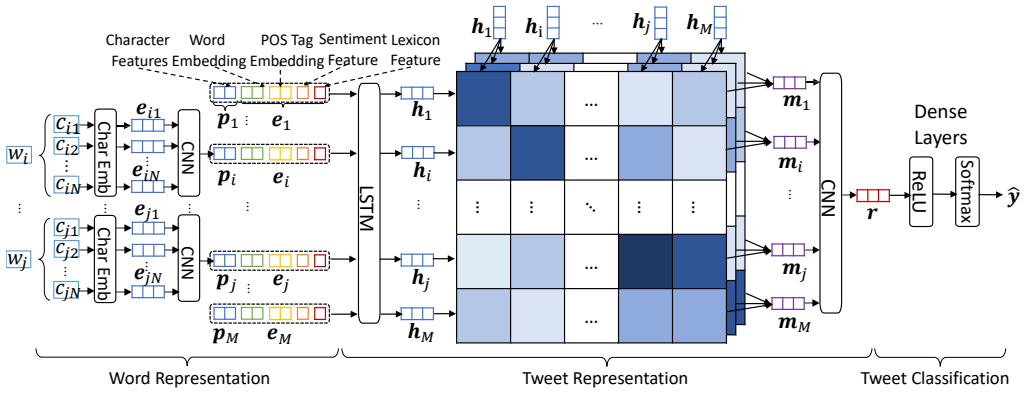

Figure 1: Architecture of our HTR-MSA model.

using a character embedding matrix. The second one is a character-level CNN network to learn contextual representations of characters. CNN is effective to capture local context information. Since many drug names contain specific character combinations (e.g., "benz" and "acid"), we apply CNN to learn contextual character representations by capturing the local information of neighbor characters. We use max-pooling operation to the feature maps generated by multiple filters in CNN to select the most significant features to build word representations based on characters.

The third one is feature concatenation, where the word representation learned from characters is concatenated with additional word features to build the final word representation vector. The first additional feature is word embeddings which are pre-trained on a large corpus and contain rich semantic information of words. According to previous studies (Sarker and Gonzalez, 2015), sentiment information and medical lexicons are very important for DN and ADR detection. Therefore, we incorporate words' sentiment scores extracted from SentiWordNet 3.0 sentiment lexicon [3] and their appearance in the SIDER 4.1 medical lexicon[4] into their representation vectors. In addition, since DN and ADR mentions usually have specific POS tags (e.g., nouns), we also incorporate the embeddings of their POS tags . The final representation vector of a word is a concatenation of its character-based representation, word embedding, POS tag embedding, sentiment scores and lexicon appearance.

2.2 Tweet Representation

The tweet representation module aims to learn the representation vectors of tweets from their words. It also contains three sub-modules.

The first one is a Bi-LSTM network (Graves and Schmidhuber, 2005). Long-distance information is very important for the detection of tweets mentioning DN and ADR. For example, the tweet "I took amoxicillin last night, but I find I'm so tired today" contains an ADR mention "tired", which has a long distance to the drug name "amoxicillin". LSTM is an effective network to capture long-distance information. We use Bi-LSTM network in our approach. It can capture the context information from both directions and output the hidden states at each position. Denote the hidden states of words in a tweet as $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_M]$, where M is sentence length.

The second sub-module is multi-head self-attention network. In most of existing attention mechanisms the attention weight of a word is computed only based on its hidden representation, and the relationships between different words in a text cannot be modeled. Usually, many DN and ADR mentions are context-dependent and the interactions between words are very important to detect the DN and ADR mentioning tweets. Self-attention is an effective way to capture the useful interactions between words in texts (Vaswani et al., 2017). In addition, a word may interact with multiple words. For example, in the tweet "I forgot to take aspirin and I'm in huge pain", the interaction of "aspirin" with "forgot" and the interaction of "aspirin" with"pain" are both important for ADR mention detection. Thus, we propose to use multi-head self-attention mechanism (Vaswani et al., 2017) to learn better hidden representations of words by modelling their interactions with multiple words. In this layer, the representation vector $\mathbf{m}_{i,j}$ of the j_{th} word learned by the i_{th} attention head is computed by a weighted summation of $\mathbf{H}$ as follows:

[3] http://sentiwordnet.isti.cnr.it/ (last access: Jul 19.)
[4] http://sideeffects.embl.de/ (last access: Jul 20.)

$$\hat{\alpha}^i_{j,k} = \mathbf{h}_j^T \mathbf{U}_i \mathbf{h}_k, \quad (1)$$

$$\alpha^i_{j,k} = \frac{\exp(\hat{\alpha}^i_{j,k})}{\sum_{m=1}^{M} \hat{\alpha}^i_{j,m}}, \quad (2)$$

$$\mathbf{m}_{i,j} = \mathbf{W}_i(\sum_{m=1}^{M} \alpha^i_{j,m} \mathbf{h}_m), \quad (3)$$

where $\mathbf{U}_i$ and $\mathbf{W}_i$ are the parameters of the i_{th} self-attention head, and $\alpha^i_{j,k}$ represents the relative importance of the interaction between the j_{th} and k_{th} words. In this way, the representation of each word is learned by utilizing the hidden representations of all words in the same text and modeling the interactions between this word with all other words. The multi-head representation $\mathbf{m}_j$ of the j_{th} word is the concatenation of the outputs from h different self-attention heads, i.e., $\mathbf{m}_j = [\mathbf{m}_{1,j}; \mathbf{m}_{2,j}; ...; \mathbf{m}_{h,j}]$.

The third sub-module is a word-level CNN network with max-pooling operation. Since many drug names contain specific word combinations (e.g., salicylic acid and acetic acid), local contextual information between words is important for DN and ADR detection. We apply CNN to the sequence of hidden representations of words in each tweet, and the final representation vector of a tweet $\mathbf{r}$ is obtained from the results of max-pooling on the CNN feature maps.

2.3 Tweet Classification

The tweet classification module is used to classify whether a tweet mentions DN or ADR. It contains two dense layers with ReLU and softmax activation functions respectively. The predicted label $\hat{y}$ of a tweet is computed as:

$$\mathbf{r}' = ReLU(\mathbf{U}_1 \mathbf{r} + \mathbf{b}_1), \quad (4)$$

$$\hat{\mathbf{y}} = softmax(\mathbf{U}_2 \mathbf{r}' + \mathbf{b}_2), \quad (5)$$

where $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$ are the parameters for DN and ADR mention classification. The loss function $\mathcal{L}$ used for model training is crossentropy:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_k \log(\hat{y}_k), \quad (6)$$

where $y_{i,k}$ and $\hat{y}_{i,k}$ are gold label and predicted label for the i_{th} tweet in the k_{th} label category. N is the number of labeled tweets.

3 Experiments

3.1 Datasets and Experimental Settings

The datasets provided by Task 1 and Task 3 in the shared tasks of the third SMM4H workshop (Weissenbacher et al., 2018) were used in our experiments. The first one is for detection of tweets mentioning DNs (denoted as DN). It contains 9,622 tweet IDs (4,975 positive and 4,647 negative samples) for training, and 5,382 tweet for test. The third one is for detection of tweets mentioning ADRs (denoted as ADR). It contains 25,598 tweet IDs (2,223 positive and 23,375 negative samples) for training, and 5,000 tweets for test. Since many tweets are not available now, we only crawled 9,065 and 16,694 tweets for training in DN and ADR respectively using these IDs.

In our experiments, we use the 400-dim pretrained word embeddings released by Godin et al. (2015). The Bi-LSTM network has 2×200 units. The CNN network has 400 filters with window size of 3. There are 16 heads in the multi-head self-attention network, and the output dimension of each head is 16. RMSProp is selected as the optimizer. Since the negative samples are dominant in the ADR dataset, we use the over-sampling strategy (Weiss et al., 2007) to balance the number of positive and negative samples. Besides, in order to further improve the performance of our approach, we incorporate the ensemble strategy by independently training our model for 10 times and using the average prediction results. The performance metric is F-score on positive samples.

3.2 Performance Evaluation

In this section, we evaluate the performance of our approach by comparing it with baseline methods, including: (1) SVM, support vector machine with word unigram features (Sarker and Gonzalez, 2015); (2) CNN, convolutional neural network (Huynh et al., 2016); (3) LSTM, Bi-LSTM network (Huynh et al., 2016); (4) CRNN, combining CNN and LSTM (Huynh et al., 2016); (5) RCNN, combining LSTM and CNN (Huynh et al., 2016); (6) HTR, our basic hierarchical tweet representation model without self-attention; (7) HTR-MSA, our hierarchical tweet representation model with multi-head self-attention; (8) HTR-MSA-ens, using an ensemble of our HTR-MSA models. For fair comparisons, we use the same additional word features with our approach in all baseline methods. We conducted 10-fold cross-validation on the labeled tweets and the results are summarized in Table 1. According to Table 1, our approach can outperform all the baseline methods. This may be because in our approach we learn word representations from not only the word embeddings but also the characters in words.

Method	DN	ADR
SVM	88.20	47.20
CNN	89.16	48.56
LSTM	88.78	48.28
CRNN	89.10	48.44
RCNN	89.31	48.75
HTR	89.80	49.49
HTR-MSA	90.57	50.55
HTR-MSA-ens	**91.85**	**52.48**
HTR-MSA-ens*	**91.83**	**52.20**

Table 1: The performance of different methods in the DN and ADR detection task. *Results on the test set.

Thus, our approach can be more robust to the massive misspellings of drug names in tweets and can mitigate the influence of out-of-vocabulary words. In addition, by comparing the results of HTR-MSA and HTR, we find that the multi-head self-attention network is helpful to improve the performance of our approach. This may be because the global context information is very important for detecting tweets mentioning DNs and ADRs and the multi-head self-attention network can effectively capture the interactions between words within a tweet. Besides, ensemble strategy can further improve the performance of our approach. It indicates that a more robust system can be built for detecting tweets mentioning drug names and adverse drug reactions using the ensemble of multiple models independently trained using our approach.

3.3 Influence of Additional Word Features

In this section, we conducted experiments to explore the effectiveness of additional word features and the results are shown in Table 2. According to Table 2, each kind of additional word feature, such as word embedding, POS tag embedding, sentiment score and medial lexicon features, is effective to improve the performance of our approach. In addition, among these additional word features word embedding seems to be most useful. This is probably because that pre-trained word embeddings can provide rich semantic information of words, which is important for detecting tweets mentioning DNs and ADRs.

Feature	DN	ADR
All	90.57	50.55
-Word embedding	86.45	46.29
-POS tag embedding	90.26	50.31
-Sentiment scores	90.33	50.29
-Lexicon feature	89.94	50.10

Table 2: Effectiveness of additional word features.

4 Conclusion

In this paper, we introduce our system participating in the first and the third shared tasks in the 3rd SMM4H workshop. We propose a neural approach with hierarchical tweet representation and multi-head self-attention to detect tweets mentioning DNs and ADRs. Our system achieved the first place in both tasks.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800402 and the National Natural Science Foundation of China under Grant U1705261, U1536207, U1536201 and U1636113.

References

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *WNUT*, pages 146–153.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Sifei Han, Tung Tran, Anthony Rios, and Ramakanth Kavuluru. 2017. Team uknlp: Detecting adrs, classifying medication intake messages, and normalizing adr mentions on twitter. In *SMM4H@ AMIA*, pages 49–53.

Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. 2016. Adverse drug reaction classification with deep neural networks. In *COLING Technical Papers*, pages 877–887.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Gary M Weiss, Kate McCarthy, and Bibi Zabar. 2007. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN*, 7:35–41.

Davy Weissenbacher, Abeed Sarker, Michael Paul, and Graciela Gonzalez-Hernandez. 2018. Overview of the third social media mining for health (smm4h) shared tasks at emnlp 2018. In *EMNLP*.