Adverse Drug Extraction in Twitter Data using Convolutional Neural Network

Liliya Akhtyamova
and John Cardiff
Institute of Technology Tallaght
Dublin, Ireland
Email: liliya.akhtyamova@postgrad.ittdublin.ie
john.cardiff@it-tallaght.ie

Mikhail Alexandrov
Autonomous University of Barcelona
Barcelona, Spain
Russian Presidential Academy of National Economy and
Public Administration
Moscow, Russia
Email: MAlexandrov@mail.ru

***Abstract*—The study of health-related topics on social media has become a useful tool for the early detection of the different adverse medical conditions. In particular, it concerns cases related to the treatment of mental diseases, as the effects of medications here often prove to be unpredictable. In our research, we use convolutional neural networks (CNN) with word2vec embedding to classify user comments on Twitter. The aim of the classification is to reveal adverse drug reactions of users. The results obtained are highly promising, showing the overall usefulness of neural network algorithms in this kind of tasks.**

I. Introduction

According to the Agency for Healthcare Research and Quality over 770,000 people are injured or die each year in hospitals from adverse drug reactions [1] (ADRs), making the early detection of them crucial. The pre-approval clinical trials are not fully able to get access to all consequences of actions required to detect possible ADR effects. Thus, society needs other ways of detecting side-effects of the medications, in particular those whose effects can be rather controversial, such as antidepressants, monoamine oxidase inhibitors (MAOIs), etc. Additionally, ADRs can have considerable economic and clinical costs as they often lead to hospital admission, prolongation of hospital stay and emergency department visits [2], [3]. It has been observed that about 5.3% of hospital admissions are associated with ADRs [4].

One of the methods of early detection of such events is to examine social media commentaries. Examples include the prediction of whether people will stay on or leave health forums (such as DailyStrength[1] and HealthBoards[2]) and investigating why they do so [5]. This has shown to be a promising area, as continued participation in such kind of forums can be very fruitful for both patients and doctors. Other examples include utilizing smoking cessation patterns on Facebook [6], revealing drug abuse [7] and monitoring malpractice [8] on Twitter. Moreover, social media can provide researchers with specific kinds of information that is usually unavailable due to data protection legislation, including a persons age, nationality, gender, and geolocation. It also helps to reveal users habits and interests, all of which can play a part in diagnostics and early detection of the different health disorders.

In this work we have employed different combinations of ADR dataset from Diegolab[3] augmented by the dataset for sentiment analysis classification task from Semeval-2015[4] to see how adding more data can help improve the model. The CNN algorithm used in [9] was chosen due to its novelty and the best performance on the relevant subtasks of Semeval-2015. We fit this architecture to our data, undoubtedly showing the relevance of sentiment analysis in the context of binary classification of ADR. The contributions we make in this paper are as follows: (i) We show how neural networks with a small number of preprocessing steps can tackle the difficult structure of Twitter data. (ii) We compare the differences in performance of CNN algorithms over word embeddings trained with the different kind of additional Twitter data and model with pretrained Google news and Wikipedia word embeddings. The rest of the paper is organized as follows. We provide an overview of related work in Section 2 and discuss our approach in detail in Sections 3, 4. In Section 5 we present evaluations of the performance of our approach and discuss the contribution of additional data set. We conclude the paper in Section 6, and discuss potential future work.

II. Problem Setting and Related Work

The ADR classification task is a binary task with the positive class assigned as those tweets which mention adverse side effects. Although some aspects from sentiment analysis classification could be applicable here, it should be understood that these two tasks are not the same. The ADR detection task is much more difficult for NLP, due to the following reasons [10]:

1) There are frequent misspellings in the names of drugs and conditions (e.g., effexer, seroquil) and ambiguous terms for expressing adverse reactions (e.g., ruined my life, learned that the hard way). So, it is not always possible for the system to catch the pattern cause-result.

[1]https://www.dailystrength.org/
[2]http://www.healthboards.com/
[3]http://diego.asu.edu/Publications/ADRClassify.html
[4]http://alt.qcri.org/semeval2015

- "Baek suddenly losing his glow :(nd im losing my abilify to speak"
- "adderal reeeeeeallllllly helped my depression but I had terrible s/e's :(Do you have Hypothyroidism?"

2) A single post can contain both positive and negative experience of the use of drugs.
   - "I loved effexor for anxiety and depression but it raised my blood pressure too much so I had to stop'
3) The post can express negative drug-drug interaction, but not the side effect of the drug.
   - "Sertraline Buspirone Lexapro and Abilify really messed up. I felt like Theon Greyjoy :("
4) Tweets also can be about drug abuse.
   - "I'm in pain. I mixed my antibiotics with my lexapro, and now I feel like I have the flu. :("
5) The ADR effects mentioned in a post may not reflect the personal experience of a writer, but refer, for example, to his overall knowledge about possible side effects or just an excerpt from the official prescription to the drug or from some other research pharmacovigilance observations.
   - "apparently itching/rash can be a side effect of wellbutrin that doesn't show up for a while after u start taking it? This is fine:("
   - "copaxone injections in the next week or so, got my health insurance sorted thankfully. Kinda nervous about the side effects"
6) The bad medical condition can be the cause of taking/not taking the medication, but not the result.
   - "not sure id be so brave with the heights! I'm not bad, struggling with appetite, pain and bloating :(may have to dbl humira."
   - "okay I only have 2 pain pills left :(no more lexapro , my knee hurts . :/"

The properties of the texts make it difficult to identify and generalize the lexical features of different posts, leading to poor performance of automatic rule-based and learning-based approaches [11]. The fact that the posts are generally very short additionally restricts the rich feature extraction via shallow processing. These factors in addition to the relatively small number of ADR related posts in the DiegoLab corpus lead to the investigation of alternative methods to tackle these problems.

CNNs with their ability of extracting a set of discriminating features at multiple levels of abstraction seemed to be a promising technology in the restricted domain of medical text processing.

There is a number of papers, which contribute to the problem of ADR detection[12], [13] A recent challenge organized by the DiegoLab research lab[5] aimed to tackle the deficit of research on the topic of ADR detection in medicine. The authors constructed a dataset using Twitter data, and labeled more than 10,000 tweets which contained the names of the 74 top selling drugs in 2013, including their misspellings. Approximately 20% of these mentioned ADRs. The unbalanced nature of the dataset became the main problem for the teams participating in this competition. The number of instances for each system is different because of the time of downloading the data from Twitter. The experiment of [11] was conducted by the authors of the dataset after the competition was hold, who were able to gather more data at that time; they showed the best result on this corpus. They used the stratified training and test splits unlike their previous work on the same data. It was the authors intention to make the experiment more approximate to real-life conditions, although these leads to the loss of the model performance.

They conducted their experiment, using Support Vector Machines with LibSVM implementation[6]; The results they got were ADR F-score 0.597, non-ADR F-score 0.943 and accuracy 90.1%. They used a set of different features (from sentiment analysis, polarity classification, topic modeling) with additional corpora from health forum and medical records added to train the model. With the position of deep learning models, there are a number of recent papers which demonstrate an outstanding performance of deep learning algorithms. In a paper [14] different implementations of convolutional and recurrent neural algorithms were implemented for this task. Although more sophisticated algorithms were represented, simple neural networks demonstrated the best result with 51% of ADR F-score. The number of data in this work is close to our, so we consider this work as our baseline. However, quite recently other authors got much more impressive results [15]. They used a huge additional corpus gathered corpus from the variety of biomedical sources to fed it then to the Semi-Supervised CNN. With this additional corpus and use of more sophisticated CNN they were able to get a result, which is 9.9% better than the result of Sarker et al [16].

Other directions of text analysis for pharmacovigilance include medical concept extraction task and relation extraction tasks The majority of approaches applied to these tasks are lexicon based [17], identifying ADRs and their interaction using a list of precompiled ADR mentions and different rules[18], [12], [19], [20]. While most approaches use lexicons for these tasks, some attempt to discover patterns in texts. The most powerful way of doing that is neural networks which are becoming increasingly popular, and they have also shown promising performances in medical image analysis [21], as well as in the variety of medical NLP tasks [22], [23], [24].

III. Models and Methods

In this section, we give an overview of Convolutional Neural Networks and describe the overall architecture of the proposed system.

A. Input Processing

In our task, the input to the classification model has the form of a user text post $\mathbf{T}$ that is treated as an ordered sequence

[5]http://diego.asu.edu/Publications/ADRClassify.html

[6]http://www.csie.ntu.edu.tw/?cjlin/libsvm/

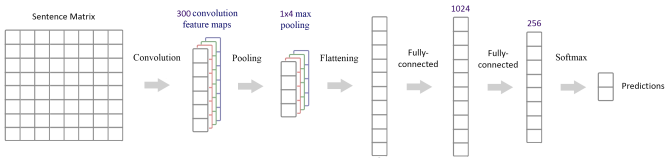

Fig. 1. Overall architecture of the proposed CNN-based model

of words $\mathbf{T} = \{w_1, w_2, ..., w_N\}$. First, plain words are mapped to their vector representations using a pre-trained word embedding model, which in our case is word2vec [25] and FastText, an instrument from Facebook, which additionally take into account the subword information [26]. The resulting representations are stacked together to form a single sentence matrix $\mathbf{M_T}$. If the original text $\mathbf{T}$ consists of N words and the dimensionality of word embeddings is d, this results in a $d \times N$ real-valued matrix which i-th column is a vector representation of the i-th word of the sentence. This matrix is then passed to CNN and further steps are described below.

B. Convolutional Neural Networks

CNN is a hierarchical feed-forward neural network which structure is inspired by the biological visual system. Its principal difference from standard neural networks is that apart from fully-connected layers it has a number of convolutional layers, where it learns filters that are sliding along the input data and applied to its sub-regions. The overall structure of CNN is described below.

a) • Convolutional layer: In one-dimensional case, a convolution between two vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f} \in \mathbb{R}^m$ is a vector $\mathbf{c} \in \mathbb{R}^{n-m+1}$, where each element c_i is computed as a scalar product between vector $\mathbf{f}$ and the correspondent subsegment of $\mathbf{x}$:

$$c_i = \mathbf{f}^T \mathbf{x}_{[i:i+m-1]}. \tag{1}$$

In other words, a vector $\mathbf{f}$, which is also called a convolutional filter, is sliding along vector $\mathbf{x}$, a dot product is computed at each step and the obtained values form the outputs of the convolutional layer. This filter is actually a parameter of CNN and its weights are learned during the training process. In two-dimensional case $\mathbf{x}$ and $\mathbf{f}$ are matrixes, and $\mathbf{f}$ is sliding not only along x-dimension, but also along y-dimension.

b) • Nonlinearity: To learn non-linear decision boundaries, convolutional layer is typically followed by non-linear activation function that is applied point-wise to its outputs. Three commonly used activation functions are *sigmoidal*, *tanh* and *ReLU*. The third one is defined as $ReLU(x) = max(0, x)$, which is a simple thresholding operation. It is used most commonly in CNNs now, and among its benefits are non-vanishing gradient in positive region and faster convergence compared to *sigmoidal* and *tanh* functions.

c) • Pooling layer: This layer usually follows a convolutional layer and its goal is to reduce and summarize the obtained representation. Two commonly used ways to do this is to take an average or maximum of small rectangular blocks of the data. Thus, if the output of the convolutional layer is a vector and the block size is k, then its size will be reduced by k times.

d) • Fully-connected layer: After several convolutional and max-pooling layers, the output of these layers, that can be treated as a new data representation, is flattened into a one-dimensional vector and used for the classification. At this stage additional external features can be added, such as bag-of-word features or averaged word embeddings. To learn non-linear dependencies, CNN has one or more fully-connected layers that perform this classification.

e) • Soft-max layer: Finally, the output $\mathbf{h}$ of the last layer is passed to soft-max function that computes probability distribution over the classes c according to the following formula:

$$p(y = c \,|\, \mathbf{h}) = \frac{e^{\mathbf{h}^T \theta_c}}{\sum_{c=1}^{K} e^{\mathbf{h}^T \theta_c}}, \tag{2}$$

where K is a number of classes and θ_c is a weight vector that corresponds to class c. This vector is also a parameter of the network that is optimized during the training.

Finally, all mentioned layers are stacked together and form one Convolutional Neural Network, that can be trained as a whole. One common way do this is to use a backpropagation algorithm and optimize training parameters with stochastic gradient descent.

C. CNN architecture

The overall architecture of the proposed CNN is presented in figure 1. It consists of one convolutional, one pooling and two fully-connected layers. The convolutional layer contains 300 filters of size $5 \times d$, where d is the dimensionality of word embeddings or the height of the sentence matrix. The number of neurons in the fully-connected layers is 1024 and 256. We use a dropout technique in these layers with dropout rate 0.2 to avoid overfitting. The CNN is trained to minimize cross-entropy loss function which is augmented with l_2-norm regularization of CNN weights. The parameters of the network are optimized with Adam modification of stochastic gradient descent using backpropagation algorithm to compute the gradients.

IV. Dataset Construction and Preprocessing

The list of URLs of partially annotated posts and a script to download them from Twitter was provided by the organizers of challenge. Due to restrictions of Twitter it was prohibited to save the posts themselves. As some people had deleted their posts we were able to download only 6929 tweets instead of the initial 10822; 749 (about 9%) of these were classified as positive. We used stratified training and test sets; 20% of data we kept for testing the model. Such proportion of the training and test data is used in other works on the same dataset [14], [16], thus we chose such split for the comparativeness purposes of our results with other works. Better constructing

of word embeddings closely depends on the amount of data fed to the model. To overcome the lack of data, we decided to use additional Twitter data to put it into the word2vec tool. The additional portion of data (200,000 and 2.5 million) tweets were added from the Subtask B on the message polarity classification task of Task 10 of the Semeval-2015 challenge. It is proven to be effective for the classification task adding additional 200,000 tweets, but not 2.5 million. We believe this happened because of the overfitting the training data with non-health related instances. Moreover, we used this additional data for the training step of our algorithm. The same technique was used in a paper of Sarker et al [16].

To make word embeddings more concise we replaced all the names of drugs with one word "drug". Additionally, we replaced all the url's mentions with the tag <url>and user names, beginning with "@", with the tag <user>. We deleted all hashtags if they didn't mention the name of a drug, to align our dataset more closely to Semeval-2015.

V. Experiments

A. Experimental Settings

Our CNN model is coded in Python and trained using TensorFlow[7], a Python instrument from Google. Hyperparameters for our model were chosen based on the ADR class F-score of the test set. Different variations on the parameters of vector representation word2vec model were tried. For our model the best result was obtained with a window of 5 and features vector size of 300, using a skipgram model. According to the training step, to handle the imbalance, we put more weight on the output of our minority class. For this purpose, we fed the data to our neural network batch in the proportion 60:40 for the positive and negative instances accordingly. Convolutional Neural Networks are trained for 20K iterations with a learning rate of 5e-4 and l_2-regularization set to 1e-2.

B. Results

The result of our CNN model implementation is shown in Table 1.

We evaluated our system using five different types of word embeddings: the original one from Diegolab corpus, which performed poorly in all variations of training data. Then, we added more data from Semeval corpus, used for the Sentiment analysis challenge. We augmented the data fed to word2vec with full sized 2.5 million tweets from this challenge, as well as with the some portion of them of 200,000. Also, we tried to put different combinations of data from two datasets in CNN. Moreover, we fed to our model GoogleNews pretrained word embeddings[8], which consist of 3 million 300-dimension English word vectors. And finally, we used Wikipedia word embeddings pretrained with the FastText toolkit [27]. It can be seen, the best result with an ADR F-score of 54.2% is obtained with the GoogleNews word vectors, although the result obtained with the use of Wikipedia corpus is slightly worse.

[7]https://www.tensorflow.org/

[8]https://github.com/mmihaltz/word2vec-GoogleNews-vectors

Training Data	Model	ADR F-score	Non-ADR F-score	Accuracy (%)
Huynh et al.[14]	CNN+glove	0.51	-	-
original	bow+logistic regression	0.367	0.851	71.0
	CNN+word2vec	0.324	0.732	61.6
	CNN+word2vec(+2.5m)	0.426	0.892	81.6
	CNN+word2vec(+0.2m)	0.483	0.936	88.6
	CNN+GoogleNews	**0.542**	0.946	90.4
	CNN+Wikipedia	**0.540**	0.942	90.2
original +0.2m	CNN+word2vec	0.301	0.687	56.7
	CNN+word2vec(+2.5m)	0.373	0.914	87.5
	CNN+word2vec(+0.2m)	0.465	0.934	88.2

TABLE I
CLASSIFICATION PERFORMANCES OVER THE ORIGINAL AND AUGMENTED DATA SETS. ADR F-SCORES, NON-ADR F-SCORES, ACCURACIES FOR EACH OF THE TRAIN-TYPES OF WORD REPRESENTATION SET COMBINATIONS ARE SHOWN. [M=MILLION]

Additionally, we conducted experiments to check the sensitivity of the model to the variation in the size of training and test sets. The results of the experiment on the GoogleNews word embeddings are presented in Table 2.

Training sample size (%)	Precision	Recall	ADR F-score
50	0.458	0.486	0.472
60	0.479	0.537	0.506

TABLE II
DEPENDENCE OF THE CNN CLASSIFIER PERFORMANCE ON TRAINING SAMPLE SIZE FOR GOOGLENEWS WORD EMBEDDINGS. PRECISION, RECALL AND F-SCORES FOR ADR-CLASS ARE SHOWN.

In this table we demonstrate the precision, recall and F1-score for ADR-class. It could be seen that with the decreasing of the training sample size by 25%, the F-score for ADR class falls only by about 6.64%.

C. Analysis of Results

While our approach did not achieve results as strong as those achieved by Diegolab team [17], the one-layer fine-tuned CNN model with pretrained Google news embeddings performs surprisingly well. However, it needs to be mentioned that we had much less available data than the Diegolab team and did not have access to the test data used in this challenge, so our results are not directly comparable. The proposed model has shown good performance by using only a small fraction of features compared to the Sarker's model with a huge number of handcrafted features gathered. Nevertheless, our system's result based on ADR F-score used as an evaluation metric in the competition outperforms the remaining teams' results on this kind of tasks and the result of the recent paper of [14], but not as good as the result of [15], who demonstrated that gathering more related data could extremely enhance the performance of the model. In the future we are planning to use the same technique, i.e. gathering more related data, to fed to our tool.

The most notable result of this work is that it is possible to obtain good results on a ADR classification task using a fast deep learning system based on already pretrained GoogleNews word embeddings – seemingly inappropriate data, gathered from Google news sources. This indicates that the size of data fed to word2vec tool had a greater impact on the outcome of our model than the quality of this data. However, we believe that if we train the model on a more relevant corpus in a size comparable to the size of the GoogleNews corpus, the results of the model will be even more impressive.

Moreover, we stated that the results of our experiment are relatively robust to the change in the proportion of the training and test samples, however, its accuracy decreases as the size of the training sample decreases. It means that we did not overestimate our model when took 80% for training and the rest for testing in our main experiment.

VI. Discussion and Future Work

In this article, we have presented a CNN based binary classification approach to the problem of ADR detection in Twitter data. In this approach, a state-of-the-art CNN architecture with one layer is implemented. Different preprocessing techniques were introduced to create better word embeddings. Finally, these embeddings are passed to a CNN to train the ADR classifier. The best result was achieved on the Google news word embeddings. We achieved an ADR F-score of 54.23% and accuracy 90.4% on the test data, showing the relevance of deep learning algorithms on this kind of tasks.

In the future, we are planning experiments which employ more intricate preprocessing, which includes adding more syntactic features, gathering additional data from Twitter to construct better word embeddings to make the training data set more balanced, as well as parsing other social media sources with ADR discussions to capture more precisely the linguistic patterns of such kind of data that could be used for the further experiments with FastText toolkit. Moreover, we intend to evaluate our model using an Ensemble of the Convolutional Neural Networks for more accurate sentence classification.

References

[1] D. C. Classen, S. L. Pestotnik, R. S. Evans, J. F. Lloyd, and J. P. Burke, "Adverse drug events in hospitalized patients. Excess length of stay, extra costs, and attributable mortality." *JAMA*, vol. 277, no. 4, pp. 301–6, 1997.

[2] J. Sultana, "Clinical and economic burden of adverse drug reactions," *Journal of Pharmacology and Pharmacotherapeutics*, vol. 4, 2013.

[3] R. Bordet, S. Gautier, H. Le Louet, B. Dupuis, and J. Caron, "Analysis of the direct cost of adverse drug reactions in hospitalised patients," *European Journal of Clinical Pharmacology*, vol. 56, no. 12, pp. 935–941, 2001.

[4] C. Kongkaew, P. R. Noyce, and D. M. Ashcroft, "Hospital Admissions Associated with Adverse Drug Reactions: A Systematic Review of Prospective Observational Studies," 2008.

[5] F. Sadeque, T. Pedersen, T. Solorio, P. Shrestha, N. Rey-Villamizar, and S. Bethard, "Why Do They Leave: Modeling Participation in Online Depression Forums," pp. 14–19, 2016.

[6] L. L. Struik and N. B. Baskerville, "The Role of Facebook in Crush the Crave, a Mobile- and Social Media-Based Smoking Cessation Intervention: Qualitative Framework Analysis of Posts," *Journal of Medical Internet Research*, vol. 16, no. 7, p. e170, 7 2014.

[7] A. Sarker, A. Nikfarjam, and G. Gonzalez, "Social Media Mining Shared Task Workshop," 2016.

[8] A. Nakhasi, R. J. Passarella, S. G. Bell, M. J. Paul, M. Dredze, and P. J. Pronovost, "Malpractice and Malcontent: Analyzing Medical Complaints in Twitter," 2012.

[9] A. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," 2015.

[10] L. Akhtyamova, M. Alexandrov, and J. Cardiff, "Review of Trends in Health Social Media Analysis," Proc. of 12-th Intern. Conf. on Computer Sciences and Information Technologies, Ed. IEEE, 2017, p. 4.

[11] A. Sarker, K. OConnor, R. Ginn, M. Scotch, K. Smith, D. Malone, and G. Gonzalez, "Social Media Mining for Toxicovigilance: Automatic Monitoring of Prescription Medication Abuse from Twitter," *Drug Safety*, vol. 39, no. 3, pp. 231–240, 2016.

[12] M. Yang, X. Wang, and M. Kiang, "Identification of Consumer Adverse Drug Reaction Messages on Social Media," *PACIS 2013 Proceedings*, 2013.

[13] J. Hadzi-Puric and J. Grmusa, "Automatic Drug Adverse Reaction Discovery from Parenting Websites Using Disproportionality Methods," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 8 2012, pp. 792–797.

[14] T. Huynh, Y. He, A. Willis, and S. Rüger, "Adverse Drug Reaction Classification With Deep Neural Networks," *Proceedings of COLING 2016: Technical Papers, COLING*, pp. 877–887.

[15] K. Lee, A. Qadir, S. A. Hasan, V. Datla, A. Prakash, J. Liu, and O. Farri, "Adverse Drug Event Detection in Tweets with Semi-Supervised Convolutional Neural Networks," in *Proceedings of the 26th International Conference on World Wide Web - WWW '17*. New York, New York, USA: ACM Press, 2017, pp. 705–714.

[16] A. Sarker and G. Gonzalez, "Portable automatic text classification for adverse drug reaction detection via multi-corpus training," *Journal of Biomedical Informatics*, vol. 53, pp. 196–207, 2015.

[17] A. Sarker, R. Ginn, A. Nikfarjam, K. O 'connor, K. Smith, S. Jayaraman, and G. Gonzalez, "Utilizing Social Media Data for Pharmacovigilance: A Review HHS Public Access," *J Biomed Inform*, vol. 54, pp. 202–212, 2015.

[18] A. Benton, L. Ungar, S. Hill, S. Hennessy, J. Mao, A. Chung, C. E. Leonard, and J. H. Holmes, "Identifying potential adverse effects using the web: A new approach to medical hypothesis generation," *Journal of Biomedical Informatics*, vol. 44, no. 6, pp. 989–996, 12 2011.

[19] I. Segura-Bedmar, R. Revert, and P. Martínez, "Detecting drugs and adverse events from Spanish health social media streams," *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi) at EACL 2014*, pp. 106–115, 2014.

[20] I. Korkontzelos, A. Nikfarjam, M. Shardlow, A. Sarker, S. Ananiadou, and G. H. Gonzalez, "Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts," *Journal of Biomedical Informatics*, vol. 62, pp. 148–158, 2016.

[21] U. K. Sikdar and B. Gambäck, "Feature-Rich Twitter Named Entity Recognition and Classification," pp. 164–170, 2016.

[22] S. K. Sahu and A. Anand, "Recurrent neural network models for disease name recognition using domain invariant features," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, no. October, pp. 2216–2225, 2016.

[23] S. Kumar Sahu, A. Anand, K. Oruganty, and M. Gattu, "Relation extraction from clinical texts using domain invariant convolutional neural network," *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, no. October, 2016.

[24] L. Akhtyamova, A. Ignatov, and J. Cardiff, "A Large-Scale CNN Ensemble for Medication Safety Analysis," *Natural Language Processing and Information Systems. NLDB 2017. Lecture Notes in Computer Science*, vol. 10260, 2017.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 1 2013.

[26] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," 7 2016.

[27] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," 7 2016.