Adverse Drug Reaction Classification With Deep Neural Networks

Trung Huynh[1,3], **Yulan He**[2], **Alistair Willis**[1] **and Stefan Rüger**[1]
[1]Knowledge Media Institute, Open University, UK
[2]Systems Analytics Research Institute, Aston University, UK
[3]Google UK
trunghlt@gmail.com, y.he@cantab.net
{alistair.willis, stefan.rueger}@open.ac.uk

Abstract

We study the problem of detecting sentences describing adverse drug reactions (ADRs) and frame the problem as binary classification. We investigate different neural network (NN) architectures for ADR classification. In particular, we propose two new neural network models, Convolutional Recurrent Neural Network (CRNN) by concatenating convolutional neural networks with recurrent neural networks, and Convolutional Neural Network with Attention (CNNA) by adding attention weights into convolutional neural networks. We evaluate various NN architectures on a Twitter dataset containing informal language and an Adverse Drug Effects (ADE) dataset constructed by sampling from MEDLINE case reports. Experimental results show that all the NN architectures outperform the traditional maximum entropy classifiers trained from n-grams with different weighting strategies considerably on both datasets. On the Twitter dataset, all the NN architectures perform similarly. But on the ADE dataset, CNN performs better than other more complex CNN variants. Nevertheless, CNNA allows the visualisation of attention weights of words when making classification decisions and hence is more appropriate for the extraction of word subsequences describing ADRs.

1 Introduction

Adverse Drug Reactions (ADRs) are potentially very dangerous to patients and are amongst the top causes of morbidity and mortality (Pirmohamed et al., 2004). Many ADRs are hard to discover as they happen to certain groups of people in certain conditions and they may take a long time to expose. Healthcare providers conduct clinical trials to discover ADRs before selling the products but normally are limited in numbers. Thus, post-market drug safety monitoring is required to help discover ADRs after the drugs are sold on the market. In the United States, Spontaneous Reporting Systems (SRSs) is the official channel supported by the Food and Drug Administration. However these system are typically under-reported and many ADRs are not recorded in the systems. Recently unstructured data such as medical reports (Gurulingappa et al., 2012b; Gurulingappa et al., 2012a) or social network data (Ginn et al., 2014; Nikfarjam et al., 2015) have been used to detect content that contains ADRs. Case reports published in the scientific biomedical literature are abundant and generated rapidly. Social networks are another source of redundant data with unstructured format. While an individual tweet or Facebook status that contains ADRs may not be clinically useful, a large volume of these data can expose serious or unknown consequences.

Common approaches to detect content with ADRs used Support Vector Machines (SVMs), Random Forest, Maximum Entropy classifiers with heavily hand-engineered features (Rastergar-Mojarad et al., 2016; Sarker et al., 2016; Zhang et al., 2016). These features normally include n-grams with different weighting schemes. When used with unigrams, these approaches suffer from the fact that their models do not take in account the interaction between terms and their orders. This problem can partially be solved by using bi-grams or trigrams. However this leads to the number of features exploding, and the models are thus easily overfitted. Meanwhile neural networks with pre-trained word representations have

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 877–887, Osaka, Japan, December 11-17 2016.

had some successes in other text classification tasks (Kalchbrenner et al., 2014; Kim, 2014; Zhou et al., 2015; Yang et al., 2016). Word representations that are typically pre-trained with unlabelled data are matrices that can be used to project words into a dense low-dimensional space (typically from 50 to 300 dimensions). These neural networks often contain convolutional filters or recurrent connections that compute weighted sums of words and their contexts.

In this paper, we train word embeddings and use them as parameters to different neural network architectures to classify documents to whether they contain ADR content. We show that even without engineered features our neural networks with word embeddings outperform maximum-entropy classifiers with different weighting schemes for n-gram features.

The rest of the paper is organised as follows. Section 2 discusses related work on ADR detection from text and briefly describes word embeddings. Various neural network architectures including two new models, Convolutional Recurrent Neural Networks (CRNN) and Convolutional Neural Network with Attention (CNNA), are presented in Section 3. Experimental setup and results are discussed in Section 4 and 5 respectively. Finally, Section 6 concludes the paper.

2 Related Work

2.1 ADR Detection from Text

Natural Language Processing (NLP) approaches have been used to detect ADRs and their relations from Electronic Health Records (EHR) (Wang et al., 2009; Friedman, 2009) and clinical reports (Aramaki et al., 2010; Gurulingappa and Fluck, 2011). Both EHRs and clinical reports have several advantages over plain text or social network data such as they contain more complete records of patients' medical history, treatments, conditions. Leaman and Wojtulewicz (2010) are ones of the first to attempt to extract ADRs from text and social networks. They generated a golden data set for DailyStrength[1], a social network where its users share health-related struggles and successes with each other, and lexicons created from UMLS Methathesaurus[2], SIDER (Kuhn et al., 2010) and The Canada Drug Adverse Reaction Database[3]. Their data set contains a total of $6,890$ comment records. Their approach is rather straightforward, which is to use direct matches of terms in their built lexicons against terms tokenised from the comments. They reported a precision of 78.3%, a recall of 69.9% and an F-score of 73.9%. Further work that focused on exploring existing or expanded lexicons to find ADRs can be found at (Benton et al., 2011; Harpaz et al., 2012; Gurulingappa et al., 2012b; Yates and Goharian, 2013; Liu and Chen, 2013). Lexicon-based approaches are limited in the number of drugs studied or the number of target ADRs. Nikfarjam and Gonzalez (2011) introduces a rule-based approach on the same DailyStrength data set. Though it does not perform as well as the lexicon-based approach, it can detect expressions not included in the lexicons.

With the emergence of annotated data, there have been more machine-learning based approaches to ADRs detection. Gurulingappa et al. (2011) used Decision Trees, Maximum Entropy and SVMs with many engineered features. They obtained an F-score of 77% for ADR class with ADE data set. Sarker and Gonzalez (2015) used SVMs with different feature sets from combined data sets (ADE, Twitter and DailyStrength). They observed that combining Twitter with ADE data sets or DailyStrength with Twitter data sets help improving their performances. Nikfarjam et al. (2015) used Conditional Random Fields to simultaneously detect ADRs and the condition for which the patient is taking the drug. In addition to traditional features, they introduced embedding clusters features trained with word2vec and k-means clustering. Rastergar-Mojarad et al. (2016) and Zhang et al. (2016) used ensemble models that combine decision trees (Random Forest) or different classifiers with various features.

Overall, approaches to ADR detection have been limited with shallow models and heavily engineered features. There has been a lack of an end-to-end approach that relies on redundancy of unannotated and annotated data.

[1] `http://www.dailystrength.org/`

[2] National Library of Medicine. 2008. UMLS Knowledge Sources.

[3] `http://www.hc-sc.gc.ca/dhp-mps/medeff/index-eng.php`

2.2 Word Embeddings

Most of deep neural networks in NLP utilise an embedding that projects each unique word into a dense lower-dimensional space (typically from 50 to 300 dimensions) and use it as the input of the network. An *embedding* is a matrix $\mathbb{R}^{\mathrm{v}\times\mathrm{s}}$, where $\mathrm{v} \in \mathbb{R}$ is the size of vocabulary and $\mathrm{s} \in \mathbb{R}$ is the number of dimensions in the low dimensional space. These embeddings are normally trained from unlabelled text that are usually redundant in huge amounts from sources like Wikipedia or CommonCrawl[4]. The embeddings are usually trained in a fashion so that the dot product of vectors of a word and its neighbour word preserves the words' point-wise mutual information (PMI) (Mikolov et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014; Shazeer et al., 2016).

After being trained, these vectors can be used to look for word synonyms by looking for words with their vectors closest to the searched word's vector. They can also be used to answer certain types of questions like "what is to Italy like Paris to France?" by looking for words with vectors that is closest to vector $\overrightarrow{\mathrm{Paris}} - \overrightarrow{\mathrm{France}} + \overrightarrow{\mathrm{Italy}} = \overrightarrow{\mathrm{Rome}}$ (Mikolov et al., 2013). By representing words using these vectors, the model captures derived information from co-occurrences of the contained words from the unsupervised pre-training. Additionally using lower dimensional vector space also helps reduce overfitting. A tokenised sentence or document with their tokens projected by an embedding becomes a dense matrix that can then be fed as an input into a neural network.

3 Methods

In this section, we introduce a number of neural network architectures and propose two new models, Convolutional Recurrent Neural Networks (CRNN) and Convolutional Neural Network with Attention (CNNA)[5].

3.1 Convolutional Neural Network (CNN)

Deep Convolutional Neural Networks (CNN)s are recently extensively used in many computer vision (Alex Krizhevsky et al., 2012; Szegedy et al., 2014; Simonyan and Zisserman, 2014; He et al., 2015) and NLP tasks. In NLP, CNNs (Figure 1a) were previously used successfully in sentence classification and sentiment analysis (Collobert et al., 2011; Kim, 2014; Zhou et al., 2015). The network starts with a convolutional layer with Rectified Linear Units (RLUs) (Glorot et al., 2011). A RLU takes an input and returns the original input if it is larger than 0, otherwise, it returns 0. The convolutional filters normally have the same width as the word vectors, thus, produce feature maps with only 1 column. The network is then stacked by a max pooling layer that picks the maximum element from each column. The last layer is a feedforward layer to an output layer with either sigmoid (Equation 3) or softmax (Equation 4) activations depending on whether the classification is binary or multinomial. The mathematical formulations for different layers of the CNN are:

$$l^k_{1_{i1}} = \max\{(W^k_1 * \mathrm{X})_{i1}, 0\}, \tag{1}$$

$$l_{2_k} = \max_i\{l^k_{1_{i1}}\}. \tag{2}$$

If it is binary classification, we set

$$l_3 = \frac{1}{1 + \exp(-W_3^\top l_2 - b_3)}, \tag{3}$$

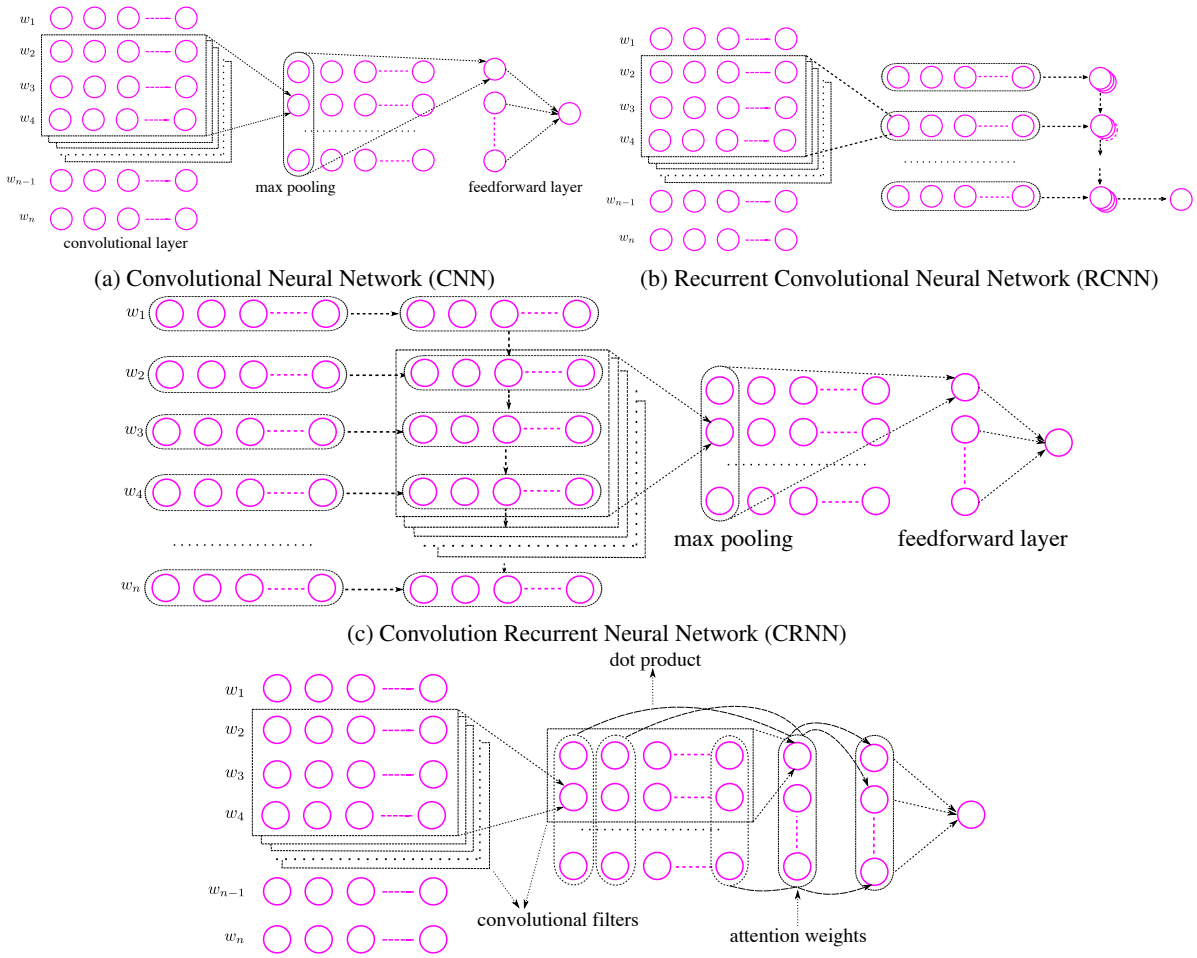
or, otherwise, if it is multinomial classification

$$l_{3_i} = \frac{\exp(W_3^\top l_2 + b_3)_i}{\sum_j \exp(W_3^\top l_2 + b_3)_j}. \tag{4}$$

Here, $\mathrm{X} \in \mathbb{R}^{\mathrm{d}\times\mathrm{s}}$ is the input matrix after the projection, $\mathrm{d} \in \mathbb{N}$ is the document length, $\mathrm{s} \in \mathbb{N}$ is the word vector length, $*$ denotes convolution, $W^i_1 \in \mathbb{R}^{\mathrm{h}\times\mathrm{e}}$, $W_3 \in \mathbb{R}^{\mathrm{k}\times 1}$ are the neural network weights, $b_3 \in \mathbb{R}$ is the bias term, $\mathrm{h} \in \mathbb{N}$ is the convolutional filter height and $\mathrm{k} \in \mathbb{M}$ is the number of convolutional filters.

[4] `http://commoncrawl.org/`

[5] Source code is available at `https://github.com/trunghlt/AdverseDrugReaction`



(a) Convolutional Neural Network (CNN)

(b) Recurrent Convolutional Neural Network (RCNN)

(c) Convolution Recurrent Neural Network (CRNN)

(d) Convolutional Neural Network with Attention (CNNA)

Figure 1: Various neural network architectures.

3.2 Recurrent Convolutional Neural Network (RCNN)

Another architecture that has achieved comparable results in sentence classification task is Recurrent Convolutional Neural Network (RCNN) (Zhou et al., 2015). The RCNN (Figure 1b) also starts with a convolutional layer like the CNN but followed by a recurrent layer rather than a max pooling layer. The convolutional filters have the same width as the embedding and are applied in the manner that the outputs have the same number of rows as the input. We also use the Rectified Linear function as the activation function for the convolutional layer. For the recurrent layer, at time step t, the recurrent node takes the input from the outputs produced by all the convolutional filters at row t and previous values at time step $t-1$. For activation, we use Gated Recurrent Units (Cho et al., 2014). Finally the nodes at the last time step are fully connected to a single node with a sigmoid activation to produce binary classification:

$$l^k_{1_{t1}} = \max\{(W^k_1 * \mathrm{X})_{t1}, 0\}, \tag{5}$$

$$l_{2_{t_j}} = \mathrm{gru}(l^*_{1_{t1}}), \tag{6}$$

$$l_3 = \frac{1}{1+\exp(-W_3^\top l_{2_\mathrm{d}} - b_3)}, \tag{7}$$

where $\mathrm{gru}(\mathrm{X}): \mathbb{R}^{d\times k} \rightarrow \mathbb{R}^{d\times r}$ denotes Gated Recurrent Unit (GRU) with input X, $r \in \mathbb{R}$ is the size of the output of the RNN, $t \in \mathbb{R}$ denotes a time step that is equivalent to the order of the window that produces the values from convolutional filters.

GRUs are recurrent units which have additional gating units. The gating units modulate the flow of information inside the unit. The activation h^i_j of a GRU at time t is a linear interpolation between

previous activations:

$$\mathrm{h}_t^j = (1 - \mathrm{z}_t^j)\mathrm{h}_{t-1}^j + \mathrm{z}_t^j \tilde{h}_t^j,$$

where z_t^j acts as a gate which decides how much the unit updates its content and it is computed by $\mathrm{z}_t^j = \sigma(W_z \mathrm{x}_t + U_z \mathrm{h}_{t-1})^j$, while $\tilde{\mathrm{h}}_t^j$ is a candidate activation, computed similarly to traditional recurrent unit, $\tilde{h}_t^j = \tanh\left(W\mathrm{x}_t + U(\mathrm{r}_t \odot \mathrm{h}_{t-1})\right)^j$, where r_t is a reset gate and $\odot$ is an element-wise multiplication. These reset gates can be computed similarly to the update gate $r_t^j = \sigma(W_r \mathrm{x}_t + U_r \mathrm{h}_{t-1})^j$.

The idea behind gated flows is to enable information further in the past to be propagated to the current unit with fewer time steps. With fewer time steps, the error gradient is passed by back-propagation more efficiently due to the propagated gradient is less prone to vanishing or exploding. (Cho et al., 2014) show that GRUs have better performance than traditional $\tanh$ and comparable performance to Long Short-Term Memory (LSTM) units.

3.3 Convolutional Recurrent Neural Network (CRNN)

Inspired by RCNN, we introduce a new architecture called Convolutional Recurrent Neural Network (Figure 1c) that stacks a convolutional layer on top of a recurrent layer, which is opposite to a RCNN. The intuition behind this is that the recurrent layer can capture the global contexts before information passed to the convolutional layer. The convolution and max-pooling layers replace the traditional average over hidden features or only hidden features at the last word in the sentence. We use GRUs for the recurrent layers and RLUs for the convolutional layer:

$$l_{1_i} = \text{gru}(\mathrm{X}_{i*}), \tag{8}$$

$$l_{2_{i1}^k} = \max\{(W_2^k * l_1)_{i1}, 0\}, \tag{9}$$

$$l_3 = \frac{1}{1 + \exp(-W_3^\top l_2 - b_3)}. \tag{10}$$

3.4 Convolutional Neural Network with Attention (CNNA)

Inspired by the works from (Bahdanau et al., 2015; Hermann et al., 2015; Rush et al., 2015; Rocktäschel et al., 2016; Yang et al., 2016) which use the attention mechanism that the generation of outputs at each consecutive time step is conditioned on different subsets of the input, we introduce a new architecture built on top of the CNN with additional attention mechanism (Figure 1d). The addition is one-filter convolutional layer on top of the direct outputs from the first convolutional layer. The outputs of this convolutional layer are normalised with *softmax* function so that they can have a sum of 1, which we call *attention weights*. These *attention weights* are then multiplied with the outputs from the first convolution (*dot product*). The outputs of this dot product are forward connected to a perceptron for binary classification.

The advantage of introducing the attention mechanism is that we can use these attention weights to extract words that the model mainly uses for the prediction. In practice, we found it very interesting and helpful to see which words are more weighted in the model's decisions (see Figure 2 in Section 5).

Even though getting more popular, attention mechanism has been mostly applied with recurrent neural networks (Bahdanau et al., 2015; Hermann et al., 2015; Rush et al., 2015; Rocktäschel et al., 2016; Yang et al., 2016). There are recently some works that incorporate attention mechanism with CNNs (Yin et al., 2016; Yin et al. , 2016). In (Yin et al., 2016), attention weights are computed differently by taking the dot product between the representation of the input query and the sentences in question-answer tasks. In (Yin et al. , 2016), even though called attention, the attention layers behave more like feature maps than traditional attention weights (multiplied with features) and are computed by matching two feature maps.

4 Experimental Setup

4.1 Datasets

We use two datasets for the evaluation of various neural network architectures. The first one is a Twitter dataset (Sarker et al., 2016) published for a shared task in Pacific Symposium on Biocomputing, Hawaii,

2016. The tweets associated with the data were collected using generic and brand names of the drugs, and also their possible phonetic misspellings. The tweets were annotated for presence of ADRs. In the shared task, 70% $(7,575)$ of the original data set is shared for training and the rest of the data is used for evaluation. Owing to Twitter's data terms and conditions, only the tweet ids are contained in the original file. At the time of this experiment, we could download only $5,108$ tweets (with 557 tweets with ADR descriptions) as many tweets are no longer accessible. Due to the difference in the size of the experimental data set, we can not compare our results directly with the previously reported baselines. Thus we reuse the codes published by (Zhang et al., 2016) that perform classification with the various algorithms (see Section 4.2 for further details).

The second dataset, the ADE (adverse drug effect) corpus, was created by (Gurulingappa et al., 2012b) by sampling from MEDLINE case reports[6]. Each case report provides important information about symptoms, signs, diagnosis, treatment and follow-up of individual patients. The ADE corpus contains $2,972$ documents with $20,967$ sentences. Out of which, $4,272$ sentences are annotated with names and relationships between drugs, adverse effects and dosages.

For both datasets, we use 10-stratified-fold cross-validation and report precision, recall and F-scores of various methods.

4.2 Baselines

For the Twitter dataset, it was reported from the shared task that both the best (Rastergar-Mojarad et al., 2016) and the second best (Zhang et al., 2016) approaches are classifiers with engineered features. In order to directly compare our results with the existing approaches, we have reimplemented these classifiers based on the published code by (Zhang et al., 2016) including term-matching classifier based on an ADR lexicon, maximum entropy with n-grams and TFIDF weightings or NB log-count ratio, and maximum entropy with word embeddings. We describe each of these methods below:

- *Term-matching based on an ADR lexicon* (**TM**). An existing ADR lexicon[7] is directly used for ADR detection. The lexicon contains $13,699$ terms describing side effects from COSTART, SIDER, CHV and DIEGO_Lab. A document is classified as positive if it contains a term from the lexicon.

- *Maximum-Entropy classifier with n-grams and TFIDF weightings* (**ME-TFIDF**). For a document $d \in \mathcal{D}$, an n-gram i has a weight of

$$F_i(d) = \begin{cases} \big(1 + \log(n_i(d))\big) \times \log\big(1 + \frac{|\mathcal{D}|+1}{|\{d' \in \mathcal{D} | n_i(d') > 0\}|+1}\big) & \text{if } n_i(d) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

  where $n_i(d)$ is the number of times a term i appears in document d.

- *Maximum-Entropy classifier with n-grams and NB log-count ratio* (**ME-NBLCR**). Each n-gram i has a weight of

$$f_i = \begin{cases} \log\big(\frac{1+\sum_{d:y(d)=1} n_i(d)}{\sum_{i' \in \mathcal{V}}(1+\sum_{d:y(d)=1} n_{i' \in \mathcal{V}}(d))} \times \frac{\sum_{i' \in \mathcal{V}}(1+\sum_{d:y(d)=-1} n_{i'}(d))}{1+\sum_{d:y(d)=-1} n_i(d)}\big) & \text{if } n_i(d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

  where $\mathcal{V}$ is a set of all n-grams and $y(d) \in \{1, -1\}$ is the true label of each document.

- *Maximum-Entropy classifier with mean word embeddings* (**ME-WE**). This method simply uses the average of embeddings of words in each document as their input into a maximum-entropy classifier.

For the ADE dataset, the best performance published is 0.81 in F-score using SVMs trained from a rich set of features including n-grams, UMLS semantic types and concept IDs, synset expansions, polarity indicator features, ADR lexicon matches, and topics, etc. (Sarker and Gonzalez, 2015). However, since our ME-NBLCR outperforms SVMs on ADE, we don't report the results using SVMs here.

[6] https://www.nlm.nih.gov/bsd/indexing/training/PUB_050.htm

[7] http://diego.asu.edu/downloads/publications/ADRMine/ADR_lexicon.tsv

Method	Twitter Dataset				ADE Dataset			
	Precision	**Recall**	**F1**	**AUC**	**Precision**	**Recall**	**F1**	**AUC**
TM	0.13	0.89	0.23	0.59	0.30	0.99	0.46	0.53
ME-TFIDF	0.33	0.70	0.45	0.85	0.74	0.86	0.80	0.94
ME-NBLCR	0.79	0.14	0.23	0.83	0.91	0.79	0.84	0.95
ME-WE	0.27	0.73	0.40	0.82	0.48	0.70	0.57	0.76
CNN	0.47	0.57	**0.51**	**0.88**	0.85	0.89	**0.87**	**0.97**
CRNN	0.49	0.55	**0.51**	0.87	0.82	0.86	0.84	0.96
RCNN	0.43	0.59	0.49	0.87	0.81	0.89	0.83	0.92
CNNA	0.40	0.66	0.49	0.87	0.82	0.84	0.83	0.95

Table 1: Adverse drug reaction classification results on the Twitter and ADE datasets.

4.3 Training of Neural Networks

In all the described neural network architectures in Section 3, the training algorithm is Adadelta (Zeiler, 2012) with learning rate of 1.0, decay rate (ρ) of 0.95 using library Keras[8]. The embedding is trained together with other parameters. For each fold, we split the training dataset into training and validating sets. The training stops when there is no performance improvement on the validation set after 5 consecutive epochs. The batch size is set as 50. All convolutional window has a size of 5.

5 Results

We compare the precision, recall and F-scores of the positive class (instances labeled as containing the description of adverse drug reactions) of neural network architectures with the baselines in Table 1. Since both the Twitter and ADE datasets contain imbalanced class distribution, we also report the Area Under the ROC Curve (AUC) results. It can be observed that in general, results on the ADE dataset are better than those on the Twitter dataset. This is perhaps not surprising since tweets contain a lot of ill-grammatical sentences and short forms. Simply relying on an ADR lexicon for the detection of ADRs from text gives the worst results. Among the baselines, the best performing method is ME-TFIDF on the Twitter dataset where an F-score of 0.45 and an AUC value of 0.85 are obtained. But on the ADE dataset with more formal language, ME-NBLCR gives superior results compared to ME-TFIDF with an F-score of 0.84 and an AUC value of 0.95. Training MaxEnt from aggregated word embeddings (ME-WE) outperforms the term matching method (TM), but performs worse than both ME-TFIDF and ME-NBLCR.

All the neural network architectures perform similarly on the Twitter dataset and they improve upon the best baseline method ME-TFIDF by 4-6% in F-score and 2-3% in AUC. On the ADE dataset, CNN outperforms other neural network architectures and its performance gain over ME-NBLCR is 7% in F-score and 3% in AUC. Overall, CNN gives the best results although CRNN and CNNA are quite close to CNN in terms of AUC values. It is not very straightforward to explain why CNNs are better than the recurrent architectures in our experiments. Our hypothesis is that as ADR descriptions are composed of short fragments of texts, convolutions with small windows are enough to capture necessary information for ADR classification.

Since CNNA assigns a weight to each word when making classification decision, we show in Figure 2 a visualisation of attention weights of sampled tweets from the Twitter dataset. Words with higher attention weights are highlighted with darker blue colour. We can observe that most of the highlighted words are indeed related to descriptions of adverse drug effects. For example, "neck ache" and "lower back pain" in the fifth tweet and "dry eyed" in the seventh tweet. The above results suggest that although CNNA gives slightly worse results compared to CNN for ADR classification, it presents results in a more interpretable form and could be potentially used for the extraction of word subsequences actually

[8] http://keras.io/

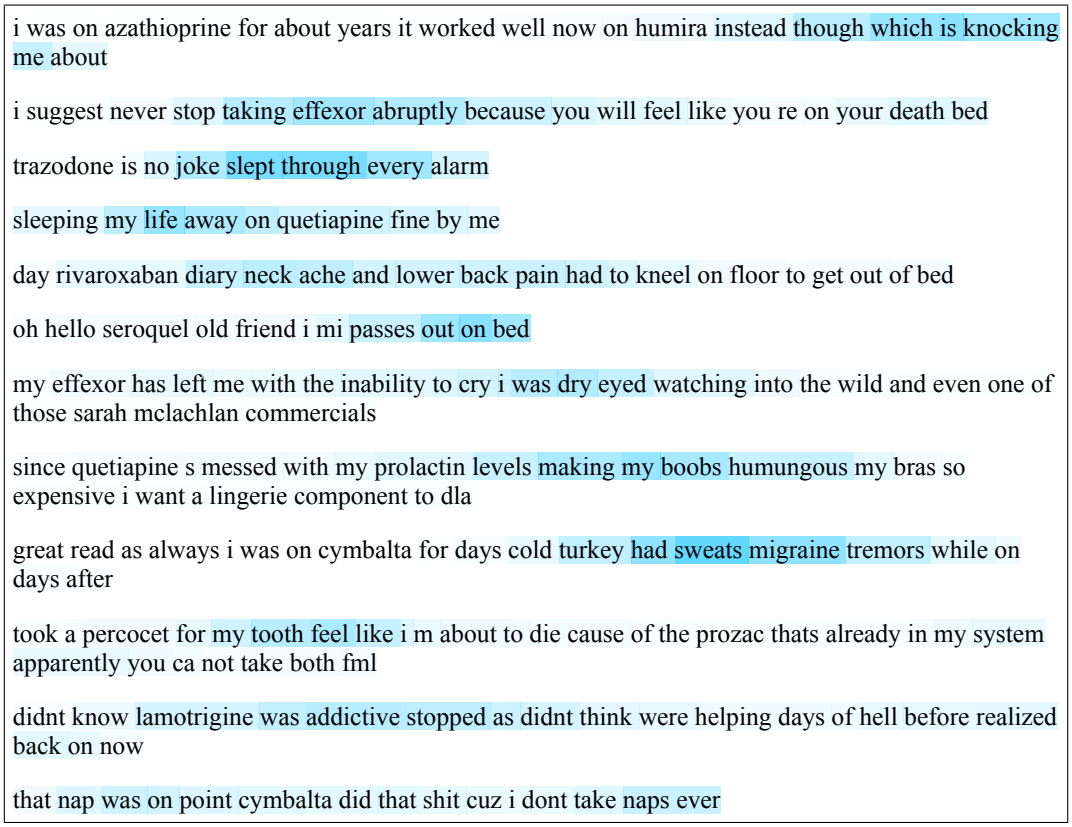


Figure 2: Sampled tweets with weighted highlights from attention weights.

describing ADRs. As such, CNNA would be a better candidate than CNN for more fine-grained ADR extraction.

6 Conclusion

This paper has explored different neural network (NN) architectures for ADR classification. In particular, it has proposed two new neural network models, Convolutional Recurrent Neural Network (CRNN) and Convolutional Neural Network with Attention (CNNA). Experimental results show that all the NN architectures outperform the traditional Maximum Entropy classifiers trained from n-grams with different weighting strategies considerably on both the Twitter and the ADE datasets. Among NN architectures, no significant differences were observed on the Twitter dataset. But CNN appears to perform better compared to other more complex CNN variants on the ADE dataset. Nevertheless, CNNA allows the visualisation of attention weights of words when making classification decisions and hence is more appropriate for the extraction of word subsequences describing ADRs.

Acknowledgements

YH is partly funded by the EPSRC AMR4AMR project (grant number EP/M02735X/1).

References

[Alex Krizhevsky et al.2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems (NIPS)*, pages 1097–1105.

[Aramaki et al.2010] Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, Kayo Waki, and Kazuhiko Ohe. 2010. Extraction of adverse drug effects from clinical records. *Studies in Health Technology and Informatics*, 160 (PART 1):739–743.

[Bahdanau et al.2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *5th International Conference on Learning Representations (ICLR)*, pages 1–15.

[Benton et al.2011] Adrian Benton, Lyle Ungar, Shawndra Hill, Sean Hennessy, Jun Mao, Annie Chung, Charles E. Leonard, and John H. Holmes. 2011. Identifying potential adverse effects using the web: A new approach to medical hypothesis generation. *Journal of Biomedical Informatics*, 44(6):989–996.

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Dzmitry Bahdanau, Holger Schwenk, Yoshua Bengio, Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empiricial Methods in Natural Language Processing (EMNLP)*.

[Collobert et al.2011] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.

[Friedman2009] Carol Friedman. 2009. Discovering Novel Adverse Drug Events Using Natural Language Processing and Mining of the Electronic Health Record. In *12th Conference on Artificial Intelligence in Medicine (AIME)*, pages 1–5.

[Ginn et al.2014] Rachel Ginn, Pranoti Pimpalkhute, Azadeh Nikfarjam, Apur Patki, Karen Oconnor, Abeed Sarker, Karen Smith, and Graciela Gonzalez. 2014. Mining Twitter for Adverse Drug Reaction Mentions: A Corpus and Classification Benchmark. In *proceedings of the 4th Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (BioTxtM)*.

[Glorot et al.2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. In *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323.

[Gurulingappa and Fluck2011] H Gurulingappa and J Fluck. 2011. Identification of adverse drug event assertive sentences in medical case reports. In *1st international workshop on knowledge discovery and health care management (KD-HCM) co-located at the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD)*, pages 16-27.

[Gurulingappa et al.2012a] Harsha Gurulingappa, Abdul Mateen-Rajput, and Luca Toldo. 2012a. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3:15.

[Gurulingappa et al.2012b] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012b. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892.

[Harpaz et al.2012] R Harpaz, W DuMouchel, N H Shah, D Madigan, P Ryan, and C Friedman. 2012. Novel data-mining methodologies for adverse drug event discovery and analysis. *Clinical Pharmacology & Therapeutics*, 91(6):1010–1021.

[He et al.2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Rian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*.

[Hermann et al.2015] Karl Moritz Hermann, Tomáš Kočiský,, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman and Phil Blunsom. 2015. Teach machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*.

[Kalchbrenner et al.2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (AL)*.

[Kim2014] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

[Kuhn et al.2010] Michael Kuhn, Monica Campillos, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2010. A side effect resource to capture phenotypic effects of drugs. *Molecular systems biology*, 6(343):343.

[Leaman and Wojtulewicz2010] Robert Leaman and Laura Wojtulewicz. 2010. Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the workshop on biomedical natural language processing (BioNLP)*, pages 117-125.

[Levy and Goldberg2014] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. *Neural Information Processing Systems (NIPS)*.

[Liu and Chen2013] Xiao Liu and Hsinchun Chen, 2013. AZDrugMiner: An Information Extraction System for Mining Patient-Reported Adverse Drug Events in Online Patient Forums. In *Proceedings of the International Conference on Smart Health (ICSH)*, pages 134–150.

[Mikolov et al.2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Neural Information Processing Systems (NIPS)*, pages 1–9.

[Nikfarjam and Gonzalez2011] Azadeh Nikfarjam and Graciela H Gonzalez. 2011. Pattern Mining for Extraction of mentions of Adverse Drug Reactions from User Comments. *AMIA Annual Symposium Proceedings*, 2011:1019–1026.

[Nikfarjam et al.2015] Azadeh Nikfarjam, Abeed Sarker, Karen O'Connor, Rachel Ginn, and Graciela Gonzalez. 2015. 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.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe : Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Pirmohamed et al.2004] Munir Pirmohamed, Sally James, Shaun Meakin, Chris Green, Andrew K Scott, Thomas J Walley, Keith Farrar, B Kevin Park, and Alasdair M Breckenridge. 2004. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18,820 patients. *BMJ*, 329(7456):15–19.

[Rastergar-Mojarad et al.2016] Majid Rastergar-Mojarad, Ravikumar Komandur Elayavilli, Yue Yu, and Hongfang Liu. 2016. Detecting signals in noisy data - can ensemble classifiers help identify adverse drug reaction in tweets? In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*.

[Rocktäschel et al.2016] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference in Learning Representation (ICLR)*.

[Rush et al.2015] Alexander M. Rush, Sumit Chopra, Jason Weston. 2015. A neural attention model for sentence summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*.

[Sarker and Gonzalez2015] 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.

[Sarker et al.2016] Abeed Sarker, Azadeh Nikfarjam, and Graciela Gonzalez. 2016. In Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing. pages 581–592.

[Shazeer et al.2016] Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. 2016. Swivel: Improving Embeddings by Noticing What's Missing. *arXiv preprint arXiv:1602.02215*.

[Simonyan and Zisserman2014] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*.

[Szegedy et al.2014] C Szegedy, W Liu, Y Jia, and P Sermanet. 2014. Going deeper with convolutions. *arXiv preprint arXiv: 1409.4842*.

[Wang et al.2009] Xiaoyan Wang, George Hripcsak, Marianthi Markatou, and Carol Friedman. 2009. Active Computerized Pharmacovigilance Using Natural Language Processing, Statistics, and Electronic Health Records: A Feasibility Study. *Journal of the American Medical Informatics Association*, 16(3):328–337.

[Yang et al.2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

[Yates and Goharian2013] Andrew Yates and Nazli Goharian. 2013. ADRTrace: Detecting expected and unexpected adverse drug reactions from user reviews on social media sites. In *European Conference on Information Retrieval (ECIR)*, pages 816–819.

[Zeiler2012] Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*.

[Zhang et al.2016] Zhifei Zhang, Jian-yun Nie, and Xuyao Zhang. 2016. An Ensemble Method for Binary Classification of Adverse Drug Reactions From Social Media. In *Proceedings of the Social Media Mining Shared Task Workshop at the Pacific Symposium on Biocomputing*.

[Yin et al.2016] Wenpeng Yin, Sebastian Ebert, Hinrich Schütze. 2016. Attention-Based Convolutional Neural Network for Machine Comprehension. In *Proceedings of NAACL Human Computer QA Workshop*.

[Yin et al. 2016] Wenpeng Yin, Hinrich Schütze, Bing Xiang, Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of Associations for Computational Linguistics*.

[Zhou et al.2015] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM Neural Network for Text Classification. *arXiv preprint arXiv:1511.08630*.