Improving classification of Adverse Drug Reactions through Using Sentiment Analysis and Transfer Learning

Hassan Alhuzali **Sophia Ananiadou**
National Centre for Text Mining
School of Computer Science, The University of Manchester, United Kingdom
{hassan.alhuzali, sophia.ananiadou}@manchester.ac.uk

Abstract

The availability of large-scale and real-time data on social media has motivated research into adverse drug reactions (ADRs). ADR classification helps to identify negative effects of drugs, which can guide health professionals and pharmaceutical companies in making medications safer and advocating patients' safety. Based on the observation that in social media, negative sentiment is frequently expressed towards ADRs, this study presents a neural model that combines sentiment analysis with transfer learning techniques to improve ADR detection in social media postings. Our system is firstly trained to classify sentiment in tweets concerning current affairs, using the SemEval17-task4A corpus. We then apply transfer learning to adapt the model to the task of detecting ADRs in social media postings. We show that, in combination with rich representations of words and their contexts, transfer learning is beneficial, especially given the large degree of vocabulary overlap between the current affairs posts in the SemEval17-task4A corpus and posts about ADRs. We compare our results with previous approaches, and show that our model can outperform them by up to 3% F-score.

1 Introduction

Social media generate a huge amount of data for health and are considered to be an important source of information for pharmacovigilance (Sloane et al., 2015; Harpaz et al., 2014; Kass-Hout and Alhinnawi, 2013). ADR detection from social media has attracted a large amount of interest as a source of information regarding morbidity and mortality. In this respect, social networks are an invaluable source of information, allowing us to extract and analyse ADRs from health communication threads between thousands of users in real-time.

Several ADR systems have utilised features related to the sentiment of words to boost their system performance (Wu et al., 2018; Kiritchenko et al., 2017; Alimova and Tutubalina, 2017; Korkontzelos et al., 2016; Sarker and Gonzalez, 2015). Korkontzelos et al. (2016) analyse the impact of sentiment analysis features on extracting ADR from tweets. The authors observed that users frequently express negative sentiments when tweeting/posting about ADRs and they found the use of sentiment-aware features could improve ADR sequence labelling and classification.

It may be observed that the language used to express sentiment is often common across different domains. Consider, for example, the tweet "I hate how Vyvanse makes me over think everything and it makes me angry about things that I shouldn't even be angry about". The keywords used in this tweet to express the authors negative sentiment towards an ADR, i.e., hate and anger, are not specific to ADRs, and may be used to express sentiment towards many different kinds of topics. Based on this observation, we hypothesise that we can leverage transfer learning techniques by using sentiment analysis data to boost the detection of ADRs.

Our main research contribution is a new neural model that detects ADRs by firstly learning to classify sentiment, using a publicly available corpus of Tweets that is annotated with sentiment information and then using transfer learning to adapt this classifier to the detection of ADRs in social media postings.

Our new ADR detection model firstly trains a classifier on the SemEval17-task4A data, which consists of Tweets on the subject of current affairs. This pre-trained classifier then is adapted to the task of detecting ADRs, using datasets of social media postings that are annotated according to the presence or absence of ADRs. To our knowledge, this is the first attempt to apply transfer learning

Proceedings of the BioNLP 2019 workshop, pages 339–347
Florence, Italy, August 1, 2019.

techniques to adapt a sentiment analysis classifier to the task of detecting ADRs. In contrast to previous research, we use generalised neural methods that avoid the use of hand-crafted features, since these are time-consuming to generate, and are usually domain-dependent. We also explore different fine-tuning methods, (Howard and Ruder, 2018; Felbo et al., 2017), to determine which one performs best in our scenario.

The rest of the paper is organised as follows: Section 2 provides a review of related work. Section 3 presents the two datasets used to create our model. Section 4 describes our method and model. Section 5 reports on the analysis of results while Section 6 provides some conclusions.

2 Related Work

There is a growing body of literature concerned with the detection and classification of ADRs in social media texts (Wang et al., 2018; Huynh et al., 2016; Ebrahimi et al., 2016; Liu and Chen, 2015). Recent work has employed sentiment analysis features to improve the classification of ADRs (Wu et al., 2018; Kiritchenko et al., 2017; Alimova and Tutubalina, 2017; Korkontzelos et al., 2016; Sarker and Gonzalez, 2015).

Nikfarjam et al. (2015) exploited a set of features, including context features, ADR lexicon, part of speech (POS) and negation, to enhance the performance of ADR extraction. The authors chose Conditional Random Field as their classifier (CRF). Korkontzelos et al. (2016) followed the same research hypothesis, but focused on the evaluation of sentiment analysis features as an aid to extracting ADRs, based on the correlation between negative sentiments and ADRs. Alimova and Tutubalina (2017) built a classification system for the detection of ADRs for which they used a Support Vector Machine (SVM), instead of CRF. The authors also explored different types of features, including sentiment features and demonstrated that they improved the performance of ADR identification. Wu et al. (2018) utilised a set of hand-crafted features (i.e. sentiment features learned from lexica), similar to all of the other studies introduced above. However, the main difference is that the model is based on a neural network architecture, including word and character embeddings, Convolutional neural network (CNN), Long Short-Term Memory (LSTM) and multi-head attentions. This was the best performing system in the 2018 ADRs shared-task[1], which is part of the social media mining for health workshop (SMM4H).

In contrast to the models proposed in the above studies, it is possible to leverage sentiment analysis features automatically, without relying on any hand-crafted features. One common approach is to pre-train a classifier on a corpus annotated with sentiment information and then to adapt this pre-trained classifier to the detection of ADRs. The advantage of this approach is that the target system only needs access to the pre-trained model, but not the original sentiment corpus, which can be important for storage and data regulation issues. This method has been investigated by various researchers (Devlin et al., 2018; Howard and Ruder, 2018; Felbo et al., 2017). Felbo et al. (2017) learned a rich representation for detecting sentiment, sarcasm, and emotion using millions of emojis' dataset, acquired from Twitter. They demonstrated that this approach performs well and can achieve results that are competitive with state of the art systems. Recently, Devlin et al. (2018) built a deep bidirectional representation from transformers, which can be fine-tuned to different target tasks with an additional output layer. The model, which is called "Bert", showed significant improvements for a wide array of tasks, such as text classification, textual entailment and question answering, among others.

Compared to the above approaches, our work uses a simpler network architecture and does not require any feature engineering. Furthermore, we take advantage of transfer learning techniques acquired knowledge from sentiment analysis data. Our work is motivated by Felbo et al. (2017) who constructed a pre-trained classifier on emoji's data and then adapted to sentiment and emotion detection. The full details of our architecture are described in section 4.1.

3 Data

Several datasets have been created for ADRs. Some of these are gathered from specialised social networking forums for health (Thompson et al., 2018; Sampathkumar et al., 2014; Yates and Goharian, 2013; Yang et al., 2012), while others are collected from social media (Ginn et al., 2014; Jiang and Zheng, 2013; Bian et al., 2012).

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

In this research, we chose a widely used dataset (containing postings from Twitter and DailyStrength[2]) (Nikfarjam et al., 2015) that are annotated according to the presence or absence of ADRs in each post. The authors partitioned the data into a training (75%) and test (25%) sets. We further divided the training set into a 60% for training and 40% for validation. The validation set is used to develop our model before it is evaluated on the original test set (i.e. 25% of the complete corpus). Our model is designed to perform binary classification, to determine whether or not a given tweet or post mentions an ADR. Table 1 presents the number of tweets/posts belong to each category in the three different partitions of the data. More detailed information about the datasets can be found in Korkontzelos et al. (2016) and Nikfarjam et al. (2015).

Datasets	#ADRs	#None
	Training	
DailyS.	900	417
Twitter	390	384
	Validation	
DailyS.	600	278
Twitter	260	256
	Test	
DailyS.	533	225
Twitter	236	192

Table 1: Data statistics (DailyS. = DailyStrength)

3.1 Sentiment Analysis corpus

We firstly train a sentiment analysis model on Twitter data from the SemEval17-task4A, which focuses on classifying the sentiment polarity of tweets on the subject of current affairs into predefined categories, e.g. positive, negative, and neutral. The dataset is partitioned into a training set of 50,000 tweets and a test set of 12,000 tweets (Rosenthal et al., 2017). A description of the sentiment analysis model is provided in section 4.

3.2 Preprocessing

Since Twitter data possesses specific characteristics, including informal language, misspellings, and abbreviations, we pre-process the data before applying the methods described in the next section. We use a tool that is specifically designed for the Twitter domain (Baziotis et al., 2017). The tool provides a number of different functionalities, such as tokenisation, normalisation, spelling-correction, and segmentation. We use the tool to tokenise the text, to convert words to lowercase, to correct misspellings, and to normalise user mentions, urls and repeated-characters.

4 Methods

This section discusses our model architecture, which is composed of two stages: the first stage involves building a sentiment analysis model, while the second stage adapts this model to a target task, which our case is the detection of ADRs. We describe our architectures in the following subsections.

4.1 Network Architecture

Our architecture consists of an embedding layer (Mikolov et al., 2013), a Long Short-Term Memory (LSTM) layer (Hochreiter and Schmidhuber, 1997), a self-attention mechanism (Bahdanau et al., 2014) and a classification layer. Figure 1 depicts the network architecture of our model.

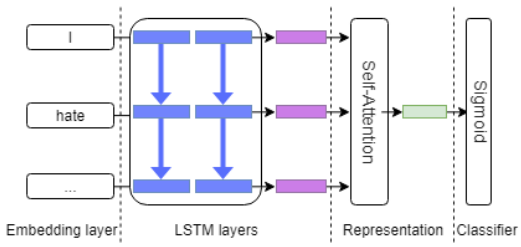

Figure 1: A description of the framework for our system.

In our different experiments, we use both an LSTM and a bi-directional LSTM (BiLSTM). Both are able to capture sequential dependencies especially in time series data, of which language can be seen as an example. The model's weights are initialized from the *word2vec* embedding with 300 dimensional size[3]. Additionally, the model consists of two LSTM/BiLSTM layers. For regularisation, we apply a dropout rate of 0.2 and 0.3 on the embedding output and after the second hidden layer, respectively, to prevent the network from over-fitting to the training set (Hinton

[2]DailyStrength is a specialised social networking website for health.

[3]https://github.com/alexandra-chron/ntua-slp-semeval2018

et al., 2012). We also choose Adam (Kingma and Ba, 2014) for optimisation and select 0.001 as the learning rate. We train the network for 10 epochs and the best performing cycle is only retained. It should be mentioned that the above set of hyper-parameters was determined using the validation set. Table 2 summarises the network architecture and hyper-parameters.

Hyper-Parameter	Value
embed-dim	300
layers	2
units	$\{200, 300, 400^*\}$
batch size	$\{32^*, 64\}$
epochs	10
sequence length	30
embed-dropout	0.2
lstm-dropout	$\{0.3, 0.4^*\}$
learning rate	0.001

Table 2: Network architecture and hyper-parameters. The asterisk (*): denotes the best performing setting

Embedding layer: T is a sequence of words $\{w_1, w_2, ..., w_n\}$ in a tweet/post and each w_i is a d dimensional word embedding for the i-th word in the sequence, where n is the number of words in the tweet. T should have the following shape n-by-d.

LSTM/Bi-LSTM layer: An LSTM layer takes as its input a sequence of word embeddings and generates word representations $\{h_1, h_2, ..., h_n\}$, where each h_i is the hidden state at time-step $_i$, retaining all the information of the sequence up to w_i. Additionally, we experiment with a BiLSTM where the vector representation is built as a concatenation of two vectors, the first running in a forward direction $\overrightarrow{h}$ from left-to-right and the second running in a backward direction $\overleftarrow{h}$ from right-to-left h_i= $[\overrightarrow{h}; \overleftarrow{h}]$.

Self-attention: A self-attention mechanism has been shown to attend to the most informative words within a sequence by assigning a weight a_i to each hidden state h_i. The representation of the whole input is computed as follows:

$$e_i = tanh(W_h h_i + b_h) \quad (1)$$

$$a_i = softmax(e_i) \quad (2)$$

$$r = \sum_{i=1}^{T} a_i \cdot h_i \quad (3)$$

, where W_h, b_h are the attention's weights.

Classification layer: The vector r is an encoded representation of the whole input text (i.e. a tweet or post), which is eventually passed to a fully-connected layer for classification. A binary classification decision is made according to whether or not the input text mentions ADRs.

Transfer Learning: There are two common approaches to transfer learning (Peters et al., 2019). One approach is to use the last layer of a pre-trained model when fine-tuning to the target task. In this scenario, the network is used as a feature extractor. An alternative approach is to use the network for initialization, i.e., the full network is unfrozen and then fine-tuned to the target task.

In this work, After training the sentiment classification model, we exclude its output layer and replace it by an ADR output layer. Finally, the network is fine-tuned to detect the ADRs adopting the same architecture and hyper-parameters as the original model. We analyse the fine-tuning methods in section 5.2.1.

5 Results & Analysis

5.1 Results

Table 3 presents the performance of our models in terms of F-score, and compares these to the three of the best performing models from recently published research. For our own results, we report the results of three different experiments. Firstly, the baseline (LSTMA) is trained to detect ADRs using only the ADR datasets mentioned above, without the use of transfer learning. The other two models (LSTMA-TL and BiLSTMA-TL) apply transfer learning, making use of pre-training of a sentiment analysis model using the SemEval17-task4A dataset. These latter two models different in terms of whether they use a single direction or bi-directional LSTM, respectively. For experiments related to previous work, we replicated the three models following their details as described in Huynh et al. (2016), Alimova and Tutubalina (2017) and Wu et al. (2018).

5.1.1 Previous Work

Alimova and Tutubalina (2017) used an SVM model with different types of hand-crafted features (i.e. sentiment and corpus-based features). Their model performed to a high degree of accuracy, which is not surprising, due to the power of the SVM model when applied to small data. Similarly,

Huynh et al. (2016) exploited different neural networks, i.e CNN and a combination of both CNN and Gated Recurrent Units (GRU). They found that CNN obtained the best performance. For this reason, the results reported in Table 3 are those obtained for the CNN model. On the Twitter dataset, the performance of the CNN is even lower than the performance of our baseline model on this dataset. However, the performance on the DailyStrength dataset is considerably higher. The model developed by Wu et al. (2018) obtained the best results among the three compared systems; indeed, the results reach the same level as our baseline system. However, it is important to note that in contrast to our model architecture, that of Wu et al. (2018) is more complex and it relies on hand-crafted features as well as deep neural architectures.

5.1.2 Contextualised Word Embedding

In this work, we also compared our model to contextualized embedding (i.e. Bert) since it has been shown to achieve high results for various NLP tasks, including text classification (Devlin et al., 2018). We use the open-source PyTorch implementations [4] and only consider the "bert-base-uncase" model. The model is trained on the default hyper-parameters except that the number of batch-size and sequence length are chosen as follows 32 and 30, respectively, to match our model hyper-parameters for these two values. As shown in Table 3, Bert model achieves the same performance as our best model "LSTMA-TL" when applied to the Twitter data, although its performance is 3% lower than our best performing model when applied to the DailyStrength dataset. Even though transfer learning is beneficial, it can achieve better performance when learned from a related domain to the problem under investigation.

5.1.3 This Work

As Table 3 demonstrates, our proposed model is able to outperform all compared systems on the DailyStrength dataset, and all systems apart from Bert when applied to the Twitter Dataset. More specifically, the "LSTMA-TL" obtained the best results, thus demonstrating the utility and advantages of transfer learning techniques. The "BiLSTMA-TL" also demonstrates competitive results for the DailyStrength dataset, but it is 1% less than the "LSTMA-TL" for the Twitter dataset. This may be due to the size of data and the architecture used in this work. Although the sentiment analysis model is trained on Twitter data, our ADR detection system still demonstrated substantial improvement on the DailyStrength dataset. Specifically, we obtained 3% and 2% improvement over our baseline model (i.e. LSTMA) on the Twitter and Dailystrength datasets, respectively.

Even though our experiments are based on a small dataset, the model demonstrated strong performance for ADR classification. Recent research claims that transfer learning techniques (i.e. fine-tuning) are beneficial for downstream tasks even if the target data size is small (Howard and Ruder, 2018; Alhuzali et al., 2018).

Datasets	**DailyS.**	**Twitter**
Models	F1	F1
Previous Work		
Huynh et al. (2016)	0.89	0.75
Alimova (2017)	0.89	0.78
Wu et al. (2018)	0.90	0.79
Contextualized W.E.		
Devlin et al. (2018)	0.89	**0.82**
This Work		
LSTMA (baseline)	0.90	0.79
LSTMA-TL	**0.92**	**0.82**
BiLSTMA-TL	**0.92**	0.81

Table 3: Comparison of our models to those reported in previous work. **LSTMA**: refers to LSTM with self−attention mechanism, while **LSTMA-TL**: means the same thing except the addition of transfer learning model. **BiLSTM-TF**: uses a BiLSTM with transfer learning model. Alimova (2017): Alimova and Tutubalina (2017). Best: bold.

5.2 Analysis

5.2.1 Impact of fine-tuning

We evaluate different methods to fine-tune our model, i.e. Last, Chain-thaw, Full and Simple Gradual unfreezing (GU). The first three techniques are adopted from Felbo et al. (2017) while the fourth one is described by Chronopoulou et al. (2019). "Last" refers to the process of only fine-tune the last layer (i.e. output layer), while the other layers are kept frozen. "Chain-thaw" method aims to firstly fine-tuned each layer independently and then fine-tuned the whole network simultaneously. "GU" is similar to the Chain-thaw method except that the fine-tuning is performed at differ-

[4] https://github.com/huggingface/pytorch-pretrained-BERT

ent epochs. In this work, we experimented with these methods and selected the one that achieved the highest results for both datasets (i.e. Twitter and DailyStrength). The results of these four methods are reported in Figure 2.

"Last", which is the standard technique in fine-tuning, achieved the lowest performance; this is not surprising, because it contains the least general knowledge. In contrast, "Chain-thaw" achieved better results than "Last". The "Full" and "GU" obtained the best results for ADR classification. When we fine-tuned the whole network, we modified the "Full" method such that the embedding layer is frozen and we called it "Full-no-Emb", instead. The intuition behind this is that the embedding layer computes a word-based representation, which does not take into account the context of a word. This method obtains the best performance for both Twitter and DailyStrength datasets.

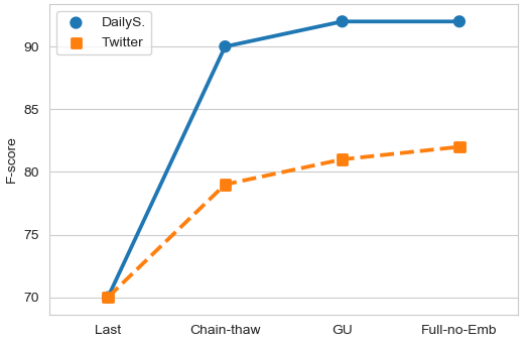

Figure 2: F-score for our model with a different set of fine-tuning methods.

5.2.2 Word Coverage

We observed that the vocabularies used in the sentiment analysis dataset and the ADR datasets share a large proportion of common words. To further investigate this, we measured the degree of common word coverage between the training and test parts of each dataset (i.e. Twitter and DailyStrength). The SemEval17-task4A training set is also included in this comparison. It should be noted that we compute the word coverage after pre-processing the data. Table 4 shows percentage of shared-vocabulary between the datasets. As shown in Table 4, the percentage of shared words between the training and test set of ADR Twitter data is 56.50%, while it is 74.22% between the SemEval17-task4A training set and the ADR Twitter test set. A similar pattern is also observed for the DailyStrength dataset, although there is a greater proportion of shared vocabulary between the training and test sets of DailyStrength. The vocabulary of the SemEval17-task4A dataset exhibits a large degree of overlap with the test sets of both Twitter and DailyStrength.

We hypothesise a number of reasons could account for this finding. Intuitively, users often use none-technical keywords when they post or tweet about ADRs. In other words, they do not employ terms found in medical lexicons. This allows users to express their opinion towards ADRs using terms which may be used to express sentiment towards other different topics. Additionally, several datasets have been collected for ADRs. However, most of them have not been made available for the research community. In contrast, there are dozens of sentiment analysis datasets available online, including SemEval17-task4A[5], Yelp reviews [6], Amazon reviews[7] and Stanford[8], among others. Thus, this confirms our initial observations and helps to reinforce that ADR system can benefit from the proliferation of sentiment analysis data available online, which is the primary motivation of this work.

Dataset	Train	SEl17-4A	Δ %
Twitter test	56.50%	74.22%	17.72%
DailyS. test	68.03%	78.22%	10.19%

Table 4: Word coverage. "SEl17-4A": corresponds to the training set of the SemEval17-task4A. Δ%: represents the difference between the two percentages for each dataset in a row.

5.2.3 Error Analysis

We experiment with small data in this work and this may limit our interpretation and analysis in this section. Nevertheless, performing error analysis can reveal some strengths and weaknesses of the proposed models and identify room for future work.

For error analysis, we selected examples which are incorrectly classified by the proposed model in this paper (i.e. LSTMA-TL) and previous work (i.e. (Huynh et al., 2016; Alimova and Tutubalina,

[5] http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools
[6] https://www.yelp.com/dataset
[7] https://s3.amazonaws.com/amazon-reviews-pds/readme.html
[8] https://nlp.stanford.edu/sentiment/index.html

2017). Figure 3 and 4 present the number of false positive and false negative classifications for each model. As can be seen in Figure 3 that the number of miss-classified examples as false negative is higher than false positive for the DailyStrength dataset, while the opposite pattern is observed for the Twitter dataset as shown in Figure 4. Our model also demonstrated balanced error classifications for both false positive and false negative. In contrast, the other two models, proposed by previous research, obtained unbalanced error classifications except Alimova and Tutubalina (2017)'s model achieved quite balanced errors for the Twitter dataset. For future work, it might be useful to investigate different ensemble methods that can help to reduce the false positive and false negative classifications and improve the classification of ADR.

In addition, we analysed examples only classified correctly by our model. We observed that our model is able to classify examples carrying none-specific keywords to ADRs, but to sentiments in general. This shows the importance of sentiment features to ADRs. Examples 1-3 below illustrate the instances that are correctly predicted by our proposed model. The first two examples are part of the Twitter test set, while the third example is part of the DailyStrength test set.

- Example 1: is it hot in here or is [durg_name] just kicking in?.
- Example 2: anyone ever taken [durg_name]? i've been on it for a week, not too sure how i feel about it yet. anyone want to share their experience?.
- Example 3: loved it , except for not being able to be woken up at night . . yeah that blew.

On the other hand, we inspected examples that our model failed to correctly classify. For instance, example (4) below was extracted from the Twitter test set and it was predicted as negative for the presence of ADR, whereas the true label is positive for the presence of ADR. Examples (5) also illustrates the same observation, but is part of the DailyStrength test set. We anticipate that our model failed to classify example (4) and (5) due to the lack of context and unambiguous keywords. Example (4) can also be interpreted as either positive or negative for the presence of ADRs. This may explain that the true label can be sometimes misleading and requires further examination.

- Example 4: moved on to something else when it quit working.
- Example 5: i'm with you. even though the [durg_name] works, i still don't feel fully human.

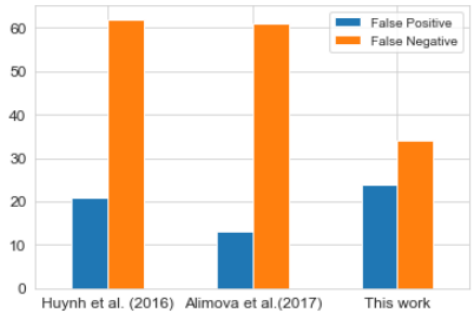

Figure 3: The number of miss-classified examples by the proposed models of this work and previous research for the DailyStrength dataset. This work: refers to the proposed model in this paper (i.e. LSTMA-TL).

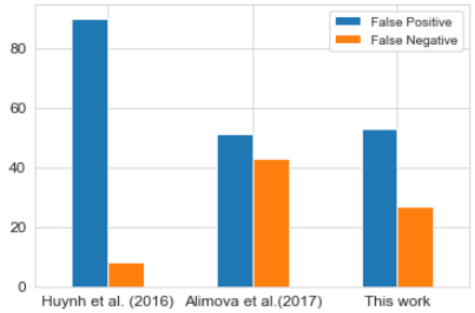

Figure 4: The number of miss-classified examples by the proposed models of this work and previous research for the Twitter dataset. This work: refers to the proposed model in this paper (i.e. LSTMA-TL).

6 Conclusion

In this work, we proposed a novel neural network architecture for ADR identification. Our approach exploits the fact that in social media, ADRs are frequently expressed with negative sentiment. Taking advantage of the readily available sentiment analysis datasets that are available online, our architecture firstly trains a sentiment analysis classifier on Tweets concerned with current affairs, and then adapts this to detect ADRs in social media. Our empirical results have demonstrated that the application of the fine-tuned model to ADR datasets obtains a substantial improvement

over previously published models. It also achieved higher results than Bert on DailyStrength dataset. Additionally, the word coverage analyses revealed that sentiment analysis dataset shares a significant amount of vocabulary with ADR dataset, which is even higher than the correlation between the words in training and test sets of the same ADR dataset. This paper has empirically discussed the advantages and utility of both sentiment analysis datasets and transfer learning techniques for improving the performance of ADR detection in social media and specialised health-related forums. Finally, we provided some error analyses and potential future work.

7 Acknowledgement

We thank Prof. Graciela Gonzalez-Hernandez, University of Pennsylvania, for sharing the Twitter and DailyStrength datasets with us. We would like also to thank Paul Thompson for his valuable comments and suggestions. The first author is supported by the Ministry of Higher Education of the Kingdom of Saudi Arabia.

References

Hassan Alhuzali, Mohamed Elaraby, and Muhammad Abdul-Mageed. 2018. Ubc-nlp at iest 2018: Learning implicit emotion with an ensemble of language models. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 342–347.

Ilseyar Alimova and Elena Tutubalina. 2017. Automated detection of adverse drug reactions from social media posts with machine learning. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 3–15. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

Jiang Bian, Umit Topaloglu, and Fan Yu. 2012. Towards large-scale twitter mining for drug-related adverse events. In *Proceedings of the 2012 international workshop on Smart health and wellbeing*, pages 25–32. ACM.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Monireh Ebrahimi, Amir Hossein Yazdavar, Naomie Salim, and Safaa Eltyeb. 2016. Recognition of side effects as implicit-opinion words in drug reviews. *Online Information Review*, 40(7):1018–1032.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

Rachel Ginn, Pranoti Pimpalkhute, Azadeh Nikfarjam, Apurv 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 fourth workshop on building and evaluating resources for health and biomedical text processing*, pages 1–8. Citeseer.

Rave Harpaz, Alison Callahan, Suzanne Tamang, Yen Low, David Odgers, Sam Finlayson, Kenneth Jung, Paea LePendu, and Nigam H. Shah. 2014. Text mining for adverse drug events: the promise, challenges, and state of the art. *Drug Safety*, 37(10):777–790.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Trung Huynh, Yulan He, Alistair Willis, and Stefan Rüger. 2016. Adverse drug reaction classification with deep neural networks. Coling.

Keyuan Jiang and Yujing Zheng. 2013. Mining twitter data for potential drug effects. In *International conference on advanced data mining and applications*, pages 434–443. Springer.

Taha A Kass-Hout and Hend Alhinnawi. 2013. Social media in public health. *Br Med Bull*, 108(1):5–24.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko, Saif M Mohammad, Jason Morin, and Berry de Bruijn. 2017. Nrc-canada at smm4h shared task: classifying tweets mentioning adverse drug reactions and medication intake. *arXiv preprint arXiv:1805.04558*.

Ioannis Korkontzelos, Azadeh Nikfarjam, Matthew Shardlow, Abeed Sarker, Sophia Ananiadou, and Graciela H Gonzalez. 2016. Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. *Journal of biomedical informatics*, 62:148–158.

Xiao Liu and Hsinchun Chen. 2015. A research framework for pharmacovigilance in health social media: identification and evaluation of patient adverse drug event reports. *Journal of biomedical informatics*, 58:268–279.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Azadeh Nikfarjam, Abeed Sarker, Karen Oconnor, 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.

Matthew Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *CoRR*, abs/1903.05987.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Hariprasad Sampathkumar, Xue-wen Chen, and Bo Luo. 2014. Mining adverse drug reactions from online healthcare forums using hidden markov model. *BMC medical informatics and decision making*, 14(1):91.

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.

Richard Sloane, Orod Osanlou, David Lewis, Danushka Bollegala, Simon Maskell, and Munir Pirmohamed. 2015. Social media and pharmacovigilance: a review of the opportunities and challenges. *British journal of clinical pharmacology*, 80(4):910–920.

Paul Thompson, Sophia Daikou, Kenju Ueno, Riza Batista-Navarro, Junichi Tsujii, and Sophia Ananiadou. 2018. Annotation and detection of drug effects in text for pharmacovigilance. *Journal of cheminformatics*, 10(1):37.

C. Wang, H. Dai, F. Wang, and E. C. Su. 2018. Adverse drug reaction post classification with imbalanced classification techniques. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 5–9.

Chuhan Wu, Fangzhao Wu, Junxin Liu, Sixing Wu, Yongfeng Huang, and Xing Xie. 2018. Detecting tweets mentioning drug name and adverse drug reaction with hierarchical tweet representation and multi-head self-attention. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop and Shared Task*, pages 34–37.

Christopher C Yang, Haodong Yang, Ling Jiang, and Mi Zhang. 2012. Social media mining for drug safety signal detection. In *Proceedings of the 2012 international workshop on Smart health and wellbeing*, pages 33–40. ACM.

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*, pages 816–819. Springer.