Identifying adverse drug events mentions in tweets using attentive, collocated, and aggregated medical representation

Xinyan Zhao,[1*] **Deahan Yu,**[1*] **V.G.Vinod Vydiswaran**[2,1]
[1] School of Information, [2] Department of Learning Health Sciences, University of Michigan
{zhaoxy,deahanyu,vgvinodv}@umich.edu
* denotes equal contribution

Abstract

Identifying mentions of medical concepts in social media is challenging because of high variability in free text. In this paper, we propose a novel neural network architecture, the **C**ollocated **L**STM with **A**ttentive **P**ooling and **A**ggregated representation (**CLAPA**), that integrates a bidirectional LSTM model with attention and pooling strategy and utilizes the collocation information from training data to improve the representation of medical concepts. The collocation and aggregation layers improve the model performance on the task of identifying mentions of adverse drug events (ADE) in tweets. Using the dataset made available as part of the workshop shared task, we show that careful selection of neighborhood contexts can help uncover useful local information and improve the overall medical concept representation.

1 Introduction

Multiple studies have analyzed health forums and other social media for drug uses, pharmacovigilance, and effectiveness of medications (Nikfarjam et al., 2015; Daniulaityte et al., 2016). However, research related to drugs and adverse drug effects (ADE) in social media continues to grow rapidly. Automatically detecting ADE mentions in social media posts has been challenging due to the large variability of free text. One of the main challenges in studying natural language processing (NLP) approaches for medical information extraction is the lack of access to health-related information on social media (Weissenbacher et al., 2019).

Having a robust representation of words is important to train high-performance information extraction approaches. In domain-specific tasks, being able to properly represent domain words or concepts could significantly improve the models. While many studies have undertaken classifications of ADE mentions in posts with various state-of-the-art techniques (Nikfarjam et al., 2015; Weissenbacher et al., 2018), there is still room to improve for the task. For example, in many trained word embedding models (Pennington et al., 2014; Godin et al., 2015; Joulin et al., 2017), the embedding of each word is treated as a vector summarizing multiple semantic meanings for each word as independent dimensions. Indeed, pre-trained embeddings that are trained on a large data corpus usually provide robust representation for common words, compared to traditional feature-based techniques such as bag of words. Yet, for domain-specific tasks, a drawback of pre-trained embeddings is that representations of domain words may not be sufficiently tuned to be able to represent the expected meaning.

Attempts have been made previously to capture the word embedding for medical concepts from a variety of medical data sources (Huang et al., 2016). Similarly, domain-specific knowledge graphs have been shown effective as external resources for feature expansion to represent medical concepts (Choi et al., 2017; Wang et al., 2017). However, even domain-based knowledge graphs sometime contain redundant information stemming from how they are constructed (Yu et al., 2014; Paulheim, 2017; Zaveri et al., 2016). Following prior work by (Turenne, 2003) that show that co-occurring pattern of terms could be beneficial to classification tasks, in this work, we consider an alternate graph-based representation that utilizes local information derived from the training data set. We build a collocation graph – a word-based graph built from the training data set where nodes correspond to vocabulary words and edges between two nodes indicate the co-occurrence of the corresponding words. We investigate if a model built over the collocation graph could use

Proceedings of the 4th Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task, pages 62–70
Florence, Italy, August 2, 2019.

pre-trained word embeddings and other information to recognize medical concepts from data. We hypothesize that the representation of a medical word can be further enriched by its neighbors in the collocation graph.

In this paper, we propose **C**ollocated **L**STM with **A**ttentive **P**ooling and **A**ggregated representation (**CLAPA**), a novel approach that integrates bidirectional LSTM model with attention and pooling strategy and utilizes the collocation information in the training data set to help enhance the pre-trained word embedding of medical concepts. We show that our model leads to a significant improvement on an ADE detection task. To the best of our knowledge, this is the first attempt that utilizes local collocation information to improve the representation of domain concepts in social media.

To summarize, we make the following contributions in this paper:

- We propose a novel architecture that encodes locally stored domain information into sentence representation.
- Our work explores the possibility that limited training data could be better exploited by including attentive collocation information.
- We provide implication for other domain-related works where better representation of domain terms is important, especially when the data set is highly imbalanced.

2 Related work

Researchers have tackled the problem of identifying posts mentioning ADEs in social media in different ways. Various methods have been used in the 2018 Social Media Mining for Health Applications (SMM4H) shared task, ranging from statistical models such as support vector machines (SVM) to deep neural network models such as convolutional neural network (CNN), long short-term memory (LSTM), and bidirectional LSTM models. Fourteen teams participated in the 2018 SMM4H shared tasks (Weissenbacher et al., 2018), and used deep neural network models and various text processing steps such as correcting misspellings, accounting for class imbalance in data, and incorporating external resources. For the ADE mention classification task, the best system achieved an F1 score of 0.522, while the next best system achieved an F1 score of 0.478. The best system (Wu et al., 2018) was based on a bidirectional LSTM model with hierarchical tweet representation and multi-head self-attention.

In recent years, models such as CNN (Kim, 2014) and bidirectional LSTM (Graves and Schmidhuber, 2005) were used for text classification. In addition, models with attention mechanism, which incorporates information of other input tokens to improve representation of each token, was introduced by (Vaswani et al., 2017). Several max-pooling techniques, which help to detect important ngrams, were explored by (Jacovi et al., 2018) and (Zhou et al., 2016). Such mechanisms and technique have been powerful tools to build better text classification systems. To train distributed representations of words, (Mikolov et al., 2013) introduced Word2Vec in which each word is represented in a low-dimensional vector space. Other popular, pre-trained word embeddings include GloVe (Pennington et al., 2014), Word2vec over Twitter (Godin et al., 2015), and FastText (Joulin et al., 2017). Similarly, graph embedding techniques over large-scale networks were studied by numerous prior works, including LINE (Tang et al., 2015), DeepWalk (Perozzi et al., 2014), and Node2Vec (Grover and Leskovec, 2016). Although graph embedding is similar to word embedding, it is trained on not only nodes adjacent to each node but on the entire local network around the node. So, graph embedding could capture the relations between nodes, and has been used for multi-label classification and community detection (Grover and Leskovec, 2016; Qiu et al., 2018). Since most text-based graphs are typically reducible to a linear chain, and the ADE detection task is a binary classification problem, we focus on only the word embedding-based approaches in this paper.

3 Collocation and aggregated representation models

In this section, we describe the architecture of our model in detail. The model contains the following three key components — medical collocation embedding, sentence encoder, and max pooling. The overall architecture of our model is shown in Figure 1. For each word, the embedding is composed of two parts, namely, a pre-trained word embedding and an attentive neighborhood embedding. Attentive neighborhood embedding is de-

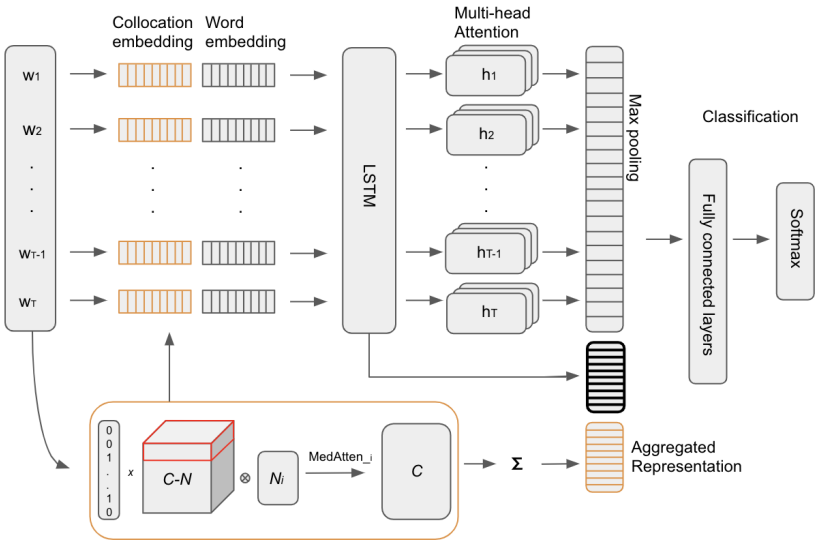


Figure 1: Overall architecture of the proposed model for identifying adverse drug events

rived from the Concept-Neighbor (C-N) tensor. In a C-N cube, each N_i represents the neighborhood for the i-th concept. Based on an attention vector ($MedAttn_i$), a concept embedding matrix C is formed in which c_i is the embedding for the concept. The collocation embedding for a word w_t will be c_i if w_t is the i-th concept, otherwise, the collocation embedding will be initialized to the zero vector. The concatenated embedding is then fed into an LSTM layer, and multi-head attention and maxpooling are applied to extract informative neurons, which are then concatenated with (1) the final state of the LSTM (sentence encoding) and (2) the sum of the concept embedding matrix. The final output is then computed via a fully connected neural network with a softmax function. Table 1 summarizes the notations used in this paper.

Notation	Definition
$W = [w_1, \ldots, w_T]$	a sequence of words
$S = [s_1, \ldots, s_{\|S\|}]$	medical concept set
$C = [c_1, \ldots, c_{\|S\|}]$	concept matrix of size $R^{\|S\| \times d}$
$N_i = [n_{i1}, \ldots, n_{iK}]$	neighborhood matrix of size $R^{K \times d}$ for the i-th concept.
C-N tensor	neighborhood tensor with the size $R^{\|C\| \times K \times d}$ composed by the neighborhood of each concept
w_t	t-th word in a text sequence.
s_i	i-th medical concept word in S
c_i	medical collocation embedding of the s_i
n_{ik}	word embedding of the k-th neighbor for the i-th concept in the concept set.
m_t	medical collocation embedding for the word w_t.
$\|S\|$	total number of concepts
T	total number of words in a sequence
K	maximum neighborhood size
L	total number of attention heads
d	dimension of word embedding
d_h	dimension of hidden states in LSTM

Table 1: Notation definitions

3.1 Medical collocation embedding

In order to better utilize the medical information embedded in text, we propose two word embedding methods – a pre-trained word embedding, and a second embedding method that enhances the pre-trained representation of medical terms by extracting information around those terms from the collocation graph.

Our medical collocation embedding can therefore be defined as following (Eq. 1):

$$\begin{aligned} MedAttn_j^i &= \frac{exp(f(n_{ij}, W_1))}{\sum_k exp(f(n_{ik}, W_1))} \\ c_i &= MedAttn_j^i \times N_i \\ m_t &= \Delta(w_t, s_i) \times c_i \end{aligned} \tag{1}$$

where $f(\cdot)$ represents a linear transformation and the $W_1^{K \times 1}$ is a trainable parameter matrix. $MedAttn_j^i$ calculates the attention that should be

paid to the j-th neighbor for the concept s_i. Therefore, the embedding c_i is represented by the embedding of its neighborhood weighted by attention scores. Lastly, m_t represents the medical collocation embedding for the t-th word in text, w_t. If the word is matched to the i-th medical concept, then $m_t = c_i$. ($\Delta(x, y) = 1$ if $x = y$; 0 otherwise).

3.2 Aggregated Medical Representation

In addition to the word-based medical concept embedding described in Sec. 3.1, we propose another aggregated medical representation strategy using the collocation information that aggregates the medical concept information in a sentence into a fixed feature space.

First, we use an attentive embedding, c_i, described in Eq. 1, to construct a medical concept representation using the neighborhood information. Then, the aggregated representation is constructed, as follows:

$$\begin{aligned} c_i^* &= c_i \oplus e(s_i) \\ \mathit{Aggre} &= \sum_i \delta(i) \times c_i^* \end{aligned} \tag{2}$$

where $e(\cdot)$ is the function that retrieves the original representation of the medical concept word from pre-trained embedding. $\delta(\cdot) = 1$, when the sentence contains the concept word, and 0 otherwise. This aggregated medical representation serves as the residual medical information that is to be added to the output layer.

3.3 Sentence encoding

To encode a sentence for the classification task, we used an attention-based LSTM to encode the entire sentence into a fixed vector space. L attention heads are applied to re-represent hidden states. The new hidden states from the l-th attention head can be described as follows (Eq. 3):

$$\begin{aligned} H, s &= \text{LSTM}([e(w_1) \oplus m_1, \ldots, e(w_T) \oplus m_T]) \\ \mathit{SentAttn}_t^l &= \frac{exp(f(h_t, W_2^l))}{\sum_k exp(f(h_k, W_2^l))} \\ \hat{h}_t^l &= \mathit{SentAttn}_t^l \cdot h_t \end{aligned} \tag{3}$$

where $H = [h_1, \ldots, h_T] \in R^{d_h \times T}$ is a hidden state matrix representing the information status at each time step, and d_h is a hidden dimension. $e(\cdot)$ and $f(\cdot)$ are the same as defined in Eq. 1. $\mathit{SentAttn}_t^l$ is a scalar representing the attention that should be paid to h_t. Therefore, $\hat{h}_t^l$ is the attentive hidden state scaled by attention values in the l-th attention head.

3.4 Max pooling layer

Motivated by previous studies (Jacovi et al., 2018; Zhou et al., 2016), the application of max pooling behavior can highlight the important signals from features and hence improve classification tasks. Following these previous approaches, we apply a max pooling layer to extract important signals from the attentive hidden state in each attention head (Eq. 4).

$$\mathit{signal}_l = \mathit{pooling}(\hat{H}_l) \tag{4}$$

where $\hat{H}_l = [\hat{h}_1^l, \ldots, \hat{h}_T^l] \in R^{d_h \times T}$, and the pooling is applied on the dimension of d_h so that $\mathit{signal}_l \in R^{d_h}$ contains important signals from each hidden dimension.

3.5 Classification layer

In the final output layer, the classification decision is made on whether or not a sentence contains an ADE mention. A fully connected network module is implemented as:

$$\begin{aligned} r &= s \oplus \mathit{signal}_1 \oplus \ldots \oplus \mathit{signal}_L \oplus \mathit{Aggre} \\ r' &= \mathit{ReLU}(U_1 r + b_1) \\ \widehat{y} &= \mathit{softmax}(U_2 r' + b_2) \end{aligned} \tag{5}$$

where r is the combination of the final state of LSTM, multiple pooled states using max pooling, and aggregated medical concept representation. Each pooled state vector signal_l comes from one attention layer (L attention layers in total) that is applied in sentence encoding (Eq. 3). U_1, U_2, b_1, and b_2 are parameters to be trained. Cross-entropy is used as the loss function for training:

$$loss = -\sum_i \sum_k y_k \log(\widehat{y}_k) \tag{6}$$

4 Experiments

4.1 Data

For our experiments, we used the data set provided as part of Task 1 of the SMM4H 2019 shared tasks (Gonzalez-Hernandez et al., 2019). As summarized in Table 2, the total number of annotated tweets is 25,678. The data set was randomly split into a training set (80%) and a validation set

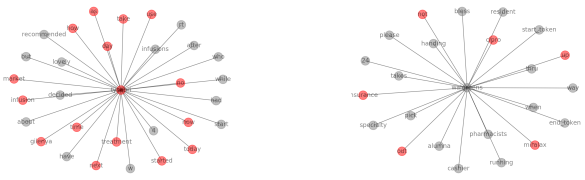
N = 25,678	**Training set** (80% of data)	**Validation set** (20% of data)
ADE tweets	1,892	485
Non-ADE tweets	18,650	4,651

Table 2: Number of ADE and non-ADE tweets in training and validation data sets.

(20%), while maintaining the target class proportions according to the original distribution. As a result, our training set contains 1,892 tweets that have an ADE mention (positive cases), and 18,650 tweets that do not have any mention of ADEs (negative cases). The validation set contains 485 positive and 4,651 negative tweets. We cleaned the tweets by separating punctuation marks, removing special characters, and replacing mentions, URLs, and number representations with normalized tokens. Finally, we used fastText (Joulin et al., 2017) as the pre-trained word embedding model.

4.2 Collocation graph

To build our collocation graph, we treat each unique word in the training set as a node, and add undirected edges from a word to adjacent words in a tweet. The collocation graph consists of 27,440 nodes and 188,329 edges. To reduce the graph size, we removed all words that appeared fewer than three times in the corpus. The resultant graph has 12,438 nodes and 159,759 edges. The mean of degree centrality is 25.39 (sd =114.59). 50% of the nodes have degrees less than 8, and 75% of the nodes have degrees less than 17.



Tysabri Walgreens

Figure 2: Examples of a collocation graph: *Tysabri* is considered as a medical concept while *Walgreens* is not considered as a medical concept.

Figure 2 shows the examples of a collocation graph. The graph has two colors: red and grey. The red nodes are words that are identified as medical concepts while the grey nodes are words that are not identified as medical concepts. The collocation graph on the left is for a medical word, *Tysabri*. The neighborhood of the word is comprised of both medical and non-medical words. *Tysabri* contains other medical words as neighbors such as *infusion*, *treatment*, and *gilenya*. The collocation graph on the right is for a word, *walgreens*. It contains few medical words such as *cipro* and *miralax*.

4.3 Medical concepts extraction

MetaMap, a widely used system for identifying medical concepts in the unified language medical system (UMLS), is used to extract potential concepts from our tweet data set (Aronson, 2006). Given a sentence as input, MetaMap identifies phrases that could be medical concepts, and maps concepts to a preferred name using UMLS. However, since MetaMap is designed to parse clinical documents rather than free text on social media, we consider only those marked phrases that are the same as the preferred name as valid medical concepts. After processing, $1,340$ concepts were extracted by MetaMap from ADE tweets and $3,921$ concepts were extracted from non-ADE tweets. Concepts are later split into single words.

4.4 Training setup

All hyperparameters are jointly trained with a learning rate of 0.001 for ten epochs. In the experiments, we used FastText pretrained embedding, and the hidden size for LSTM is set to be 300. Number of multi-head attention layer is set to be 3. For each experiment, the score is taken from the average of five runs.

4.5 Results

To evaluate our model, we set two baselines: an attention-based LSTM model (Eq. 3), and an attention-based LSTM model with max pooling (Eq. 4). The results are presented in Table 3 as rows (1) and (4), respectively.

Model	**Precision**	**Recall**	**F1**
(1). LSTM+Attn (LA)	**0.6626**	0.4495	0.5356
(2). (1)+**colloc** (CLA)	0.6392	0.4639	0.5142
(3). (2)+**Aggr** (CLAA)	0.5181	**0.5918**	**0.5525**
(4). (1)+Pool (LAP)	**0.6475**	0.4887	0.5570
(5). (4)+**colloc** (CLAP)	0.6359	0.5546	0.5925
(6). (5)+**Aggr** (CLAPA)	0.6017	**0.5979**	**0.5998**
CLAPA on Test set	**0.5944**	**0.5431**	**0.5676**
Avg. system score	0.5351	0.5054	0.5019

Table 3: Comparison of models on Precision, Recall, and F1 measures for the ADE detection task on the validation set. The scores in the last two rows are over the test set of the 2019 SMM4H 2019 shared task 1.

As presented in Table 3, the model performance is significantly improved with the addition of collocation medical embedding and aggregated embedding, over the attention-based bi-direction LSTM models. Further, adding aggregated medical information helps improve recall, but reduces the model precision and only slightly increases the F1 score, compared to the collocation based model. Hence, while highlighting medical information can reduce false negative decisions, it also causes more instances to be labeled as ADE tweets, thereby increasing a false positive rate as well. The CLAPA model, that integrates both collocation and aggregated representation along with attentive pooling strategy performs the best.

When run against the test set for the shared task, the CLAPA model achieves a F1 score of 0.5676 (see Table 3). As a comparison, the average F1 score of systems participating in this task is 0.5019. This shows our CLAPA model performs significantly better than average on this task.

4.6 Model learning stability

To show that our model consistently works better even with smaller training data, we independently and randomly sampled 10%, 30%, 50%, 70%, and 90% data from training set and retrained the models. Figure 3 shows that our model consistently performed well on the validation set, even with reduced training size, compared to the baseline model of bidirectional LSTM model with attentive pooling (the "LAP" model). The results are similar to those on the full validation data set in Table 3, in that even when only a fraction of training data is available, the model achieves higher F1 score because of significantly better recall and at a relatively small reduction in precision.

4.7 Effect of concept vocabulary

Next, we analyzed the effect of medical concepts observed in the ADE tweets to understand if there is any difference in terms of the use of medical concepts in ADE tweets vs. non-ADE tweets. We calculated a propensity ratio of each medical term, based on number of times it appears in ADE tweets compared to non-ADE tweets. We found that *causing*, *gain*, *drowsiness*, and *sweats* are likely to appear in ADE tweets about 15 times more often than in non-ADE tweets. Similarly, *crippled* is likely to appear in an ADE tweet about 26 times more often than in a non-ADE tweet. Considering the highly skewed appearance ratio

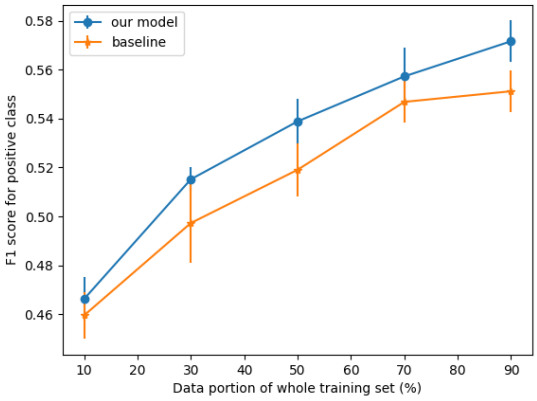


Figure 3: Effects of training size on model performance stability

for certain concepts, we analyzed the effect on using concepts from the ADE tweets alone. We compared two models – one trained over medical concepts identified from the ADE tweets and another trained over concepts from the entire training set, i.e. both ADE and non-ADE tweets.

Concepts from	Precision	Recall	F1
All tweets	0.6142	0.5546	0.5829
Only ADE tweets	0.6017	0.5979	**0.5998**

Table 4: Effects of concept vocabulary on model performance

As summarized in Table 4, the model trained with concepts from just the ADE tweets achieved a higher F1 score. While the precision is slightly lower, the model trained over concepts from ADE tweets has a significantly higher recall. On further analysis, we find that out of the $1,183$ concept words extracted from the ADE tweets, 866 concepts (73.2%) occurred more frequently in ADE tweets than in non-ADE tweets. However, when using the concepts words extracted from both ADE and non-ADE tweets, the number of concepts are higher ($n = 4,643$), but only $1,094$ concepts (23.6%) of those appear more frequently in the ADE tweets. This indicates that propensity ratio could be used for selecting medical concepts used in the ADE tweets as features.

4.8 Effects of neighborhood selection

We analyzed two additional questions related to parameter tuning:

(1) What method should be used to pick a neighbor? To answer this question, we fixed the

neighborhood size as 15 words, and selected one of the following three methods to choose neighbors:

(a) Random: Given a node n, we randomly select k of its neighbors $n_1, n_2, \ldots, n_k \in N$, where N is a set of all neighbors for node n.

(b) Popularity: For each medical concept, we first selected a neighbor that has the highest degree. When node n_i has more neighbors than node n_j, we say that node n_i is more popular than node n_j. Then, given a node n, we select k popular neighbors $n_1, n_2, \ldots, n_k$ that have the highest degree. In case of ties in popularity, neighbors are selected at random from this set.

(c) Medical neighbor: Given node n, we add k medically-related neighbors.

For all three neighborhood selection methods, if the total number of first-degree neighbors is less than k, then an additional random selection is used among second-degree neighbors to fill the gap.

Table 5 shows the results using different selection methods under the two scenarios described in Section 4.7. The left column depicts the model trained on concepts from all tweets, and the right column represents the model trained with concepts from ADE tweets alone.

	F1 scores	
Selection method	**ADE+non-ADE**	**ADE**
Random	0.5796	0.5683
Popularity	0.5819	**0.5998**
Medical neighbor	**0.5829**	0.5887

Table 5: Effects of neighborhood selection methods on F1 scores on both ADE+non-ADE tweets and only ADE tweets

Table 5 shows that targeting at neighbors using either *popularity* or *medical* attributes always leads to better performance regardless of different scenarios. However, when using medical concepts of both ADE and non-ADE tweets, picking a medical neighborhood could be a better choice, whereas popular neighborhood is preferred when concepts are identified from ADE tweets. Medical neighborhood has a higher probability of including informative words related to ADE; and when only ADE tweets are considered, the frequency of co-occurrence of a neighbor and the concepts become more important. This explanation also aligns with how language models are usually trained.

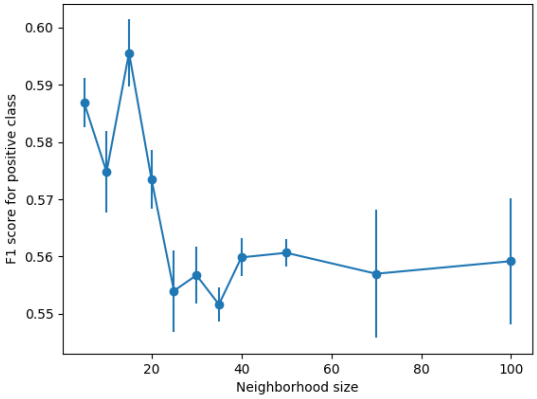


Figure 4: Effects of neighborhood size (k) on model performance

(2) How should we decide neighborhood size? We experimented with different neighborhood size. As shown in Fig. 4, as the neighborhood size k increases, the performance is not affected much when k is small (from 5 to 20). However, the performance drops significantly when k is larger ($k > 20$). We explain this by aligning back to our neighborhood selection method where we found that choosing good neighbors (popular or medically related) favors the model. We want to choose informative neighbors instead of all neighbors. Therefore, when k is small, the selected neighbors (high degree) can be easily differentiated from the ones not selected. However, when k is large, the selected neighbors become less informative because many unimportant, noisy, neighbor words (low degree/non-popular) may be included that harm the model.

5 Limitation and future work

After the above examination of our model, we argue that our model suffers from three main limitations. First, although MetaMap has been found useful at parsing medical notes, due to the different linguistic use on social media, running MetaMap on tweets may not identify relevant concepts. Second, the use of collocation graph and aggregated medical concept representation reduced precision of models, although the overall recall and F1 improved. Additional studies are need to further improve the precision. Third, the collocation graph is built solely on the training data set.

This may not favor the model when the data set is not representative enough to provide neighborhood of high quality. To address the first two issues, we believe a pre-trained state-of-the-art medication detection system could be helpful to identify high-quality medical concepts from tweets. For the third issue, we plan to use domain based knowledge base such as UMLS to expand the coverage of the limited data.

We used fastText as the pre-trained word embedding for our model. While fastText is trained on sub-word representations, models trained over medical or larger text corpora might provide additional contextual representation. Additional studies are needed to test our model on different pre-trained word embeddings such as Word2vec over Twitter (Godin et al., 2015). We also note that there is a difference in the use of medical related concepts in different classes by testing two scenarios — a model using medical concepts identified from both ADE and non-ADE cases and one using those from the ADE cases. In future, we plan to test this approach by exploring the use of unique nodes in different classes. Meanwhile, the application of our approach on other domain-specific tasks should be verified to examine the generalization of the approach.

6 Conclusion

In this work, we argue that a collocation graph can be utilized to enrich the representation of a medical concept. We further propose a novel neural network architecture that uses attentive information from a collocation graph to re-embed medical words. Our experiments show that, with a good selection of neighborhood, more useful local information can be accessed, which in turn improves the medical concept representation and the overall model performance in detecting mentions of adverse drug events in tweets.

Acknowledgment

We would like to thank Daniel Romero for his advice and feedback throughout the study.

References

Alan R Aronson. 2006. Metamap: Mapping text to the umls metathesaurus. *Bethesda, MD: NLM, NIH, DHHS*, pages 1–26.

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. 2017. Gram: Graph-based attention model for healthcare representation learning. In *KDD*.

Raminta Daniulaityte, Lu Chen, Francois R Lamy, Robert G Carlson, Krishnaprasad Thirunarayan, and Amit Sheth. 2016. "when 'bad' is 'good' ": Identifying personal communication and sentiment in drug-related tweets. *JMIR Public Health and Surveillance*, 2(2):e162.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China. Association for Computational Linguistics.

Graciela Gonzalez-Hernandez, Davy Weissenbacher, Michael Paul, Abeed Sarker, Ari Z. Klein, Arjun Magge, Ashlynn R. Daughton, and Karen O'Connor. 2019. Social media mining for health applications (smm4h) workshop & shared task 2019. Https://healthlanguageprocessing.org/smm4h/.

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

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, pages 855–864.

Jian Huang, Keyang Xu, and V.G. Vinod Vydiswaran. 2016. Analyzing multiple medical corpora using word embedding. In *IEEE International Conference on Healthcare Informatics (ICHI)*, volume 1, pages 527–533.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

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 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 : JAMIA*, 22(3):671–681.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. 2014. Deepwalk: online learning of social representations. In *KDD*.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*.

Nicolas Turenne. 2003. Learning semantic classes for improving email classification. In *Proceedings of Text Mining and Link Analysis Workshop*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Meng Wang, Mengyue Liu, Jun Liu, Sen Wang, Guodong Long, and Buyue Qian. 2017. Safe medicine recommendation via medical knowledge graph embedding. *CoRR*, abs/1710.05980.

Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael Paul, and Graciela Gonzalez-Hernandez. 2019. Overview of the fourth Social Media Mining for Health (SMM4H) shared task at ACL 2019. In *Proceedings of the 2019 ACL Workshop SMM4H: The 4th Social Media Mining for Health Applications Workshop & Shared Task*.

Davy Weissenbacher, Abeed Sarker, Michael J Paul, and Graciela Gonzalez-Hernandez. 2018. Overview of the third social media mining for health (smm4h) shared tasks at emnlp 2018. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*, pages 13–16.

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 *Association for Computational Linguistics*, The 3rd Social Media Mining for Health Applications Workshop and Shared Task.

Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare Voss, and Malik Magdon-Ismail. 2014. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1567–1578.

Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2016. Quality assessment for linked data: A survey. *Semantic Web*, 7(1):63–93.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.