Contextualized Graph Embeddings for Adverse Drug Event Detection

Ya Gao[1], Shaoxiong Ji*[1], Tongxuan Zhang[2], Prayag Tiwari[1], and Pekka Marttinen[1]

[1]*Aalto University, Espoo 02150, Finland*
[2]*Tianjin Normal University, Tianjin 300387, China*
Email: {ya.gao; shaoxiong.ji; prayag.tiwari; pekka.marttinen}@aalto.fi; txzhang@tjnu.edu.cn

Abstract

An adverse drug event (ADE) is defined as an adverse reaction resulting from improper drug use, reported in various documents such as biomedical literature, drug reviews, and user posts on social media. The recent advances in natural language processing techniques have facilitated automated ADE detection from documents. However, the contextualized information and relations among text pieces are less explored. This paper investigates contextualized language models and heterogeneous graph representations. It builds a contextualized graph embedding model for adverse drug event detection. We employ different convolutional graph neural networks and pre-trained contextualized embeddings as the building blocks. Experimental results show that our methods can improve the performance by comparing recent ADE detection models, suggesting that a text graph can capture causal relationships and dependency between different entities in a document.

Keywords— Adverse Drug Events Graph Neural Networks Contextualized Embeddings.

1 Introduction

Adverse Drug Events (ADEs) are injuries resulting from medical intervention related to a drug [8]. A typical way to detect ADEs is to conduct a clinical trial. However, there are many settings where a drug would be used, and we cannot check all of them during the clinical trial. Besides, some ADEs have long latency, making them hard to be discovered by an ordinary clinical trial [32]. Post-marketing drug safety surveillance, also called pharmacovigilance, is conducted to solve these problems. Pharmacovigilance activities mostly depend on Spontaneous Reporting Systems, which collect users' voluntary ADE reports [20]. However, the number of people willing to report their experiences through the official systems is negligible. Furthermore, these systems are limited due to biased and incomplete reports.

Compared with reports using Spontaneous Reporting Systems, more people often talk about their adverse reactions on social media platforms. Recent publications collect documents from social media such as Twitter and Reddit to obtain more reliable data and detect ADEs automatically using Nature Language Processing (NLP) techniques. The detection of ADEs can be seen as a text classification task or a sequence-labeling problem, where we need to identify documents including ADEs [9]. The early studies include lexicon-based and rule-based methods [38, 31]. These methods focus on string-matching, which is less effective for social media text and consumes many resources to build rules. Machine learning algorithms are also used to solve this task, such as Conditional Random Fields (CRFs) [26] , Support Vector Machine (SVM) [4], Recurrent Neural Network (RNNs) [5] and Convolutional Neural Networks (CNNs) [11]. These approaches can process text with manual feature engineering or enable automated feature learning with deep learning methods, facilitating automated ADE detection from biomedical text or social content. However, the existing approaches and models have two limitations: (1) some works are limited in capturing the rich context information in the text. (2) some do not fully consider the causal relationship and dependency between different entities in a document. Effective text encoding should be considered for the ADE detection task to capture rich semantic and contextualized information. Note that detecting causal relationships does not here refer to causal inference as in the field of machine learning focusing on causality [27], but rather expressing or indicating the relationship between the cause, e.g. a drug taken, and the respective individual's adverse health outcome as reported in the text sample.

*Corresponding author

Graphs are commonly used for different data representations because of their strong expressivity. Text data can be represented by heterogeneous graphs, where different words, phrases, and documents are seen as nodes, and their relations are shown using edges. Text graphs and graph neural networks are widely used in many NLP applications for healthcare tasks such as sentiment classification and review rating [39, 22]. Graph Neural Networks (GNNs) [36] can be applied to graph representation learning and capture the causal relationships and dependency of objects, making them more suitable for representing text with adverse drug events. However, no existing studies on ADE detection employ graph representation and graph neural networks. Besides, contextualized representations of text facilitate various NLP applications and boost the performance of NLP systems with minimal architecture engineering. In the medical domain, contextualized embeddings with domain knowledge are also in need. Pretrained contextualized language embeddings have been applied to various medical applications such as medical code assignment [12] and biomedical knowledge graph construction [13].

This paper presents a contextualized graph embedding model for ADE detection. We build contextualized language embeddings to capture contextualized information. With a heterogeneous graph built to embody word and document relations from the ADE corpus, we use graph neural networks to learn causal relations between word and document nodes to improve adverse drug reaction detection. This paper deploys different GNN-based models and pre-trained contextualized embeddings. The performance of these models is evaluated and compared with state-of-the-art models on three public benchmarks for ADE detection. Our model outperforms several strong ADE detection models in most cases. We also analyze the experiment results to discuss some potential challenges and explore the potential for improving the ADE detection tasks. The code will be made publicly available on acceptance.

Our contributions include the following folds.

- We develop a contextualized graph embedding model (CGEM) that introduces text graphs to capture the cause-effect relation for drug adverse event detection.
- The CGEM model utilizes contextualized embeddings pre-trained in large-scale domain-specific corpora for capturing context information, convolutional GNNs for text graph encoding, and an attention classifier for ADE classification.
- Experimental results show our approach outperforms recent advanced ADE detection models in three public datasets from the biomedical domain and social media.

2 Related Work

The rapid development of deep learning makes neural network-based approaches predominant in ADE detection. RNN can process sequence information and capture the sequential dependency, making it is suitable for ADE detection from text. Many studies on the ADE detection task employ RNN-based models. Cocos et al. [5] developed a Bidirectional Long Short-Term Memory (BiLSTM) network to label different parts of a sequence for ADE detection. Dandala et al. [6] presented a pipeline-based system to recognize entities relevant to ADEs using BiLSTM and CRF and then determine relations between different entities using BiLSTM and attention network. Information from recognition of concepts and relations can benefit each other, enabling this joint modeling technique to obtain more useful information during learning. However, inaccurate recognition in the first step will affect the following steps, known as the error propagation issue. To address this issue, Wei et al. [34] proposed a joint learning model which can recognize entities of ADE, the reason, and their relations simultaneously. In the recognition phase, the joint model employs CRF and BiLSTM. To achieve relation classification, it uses CNN-RNN and SVM.

Some studies also developed models with other neural network architectures, such as capsule networks and the self-attention mechanism. Zhang et al. [42] presented a model called Gated iterative capsule network (GICN), which applies CNN to obtain the complete phrase information and extracts deep semantic information using a capsule network with a gated iteration unit. This unit can remember contextual information by clustering features. However, they did not consider the wights of different parts of a document. With attention mechanisms, more critical parts of a document get higher weights. Kayesh et al. [16] proposed a causality-driven neural network-based model which applies a multi-head self-attention mechanism to learn word-to-word interactions. Ge et al. [10] employed Multi-Head Self-Attention in their model to distinguish the importance of different words. Wunnava et al. [37] developed a dual-attention mechanism with BiLSTM to capture both task-specific and semantic information in the sentence. However, they did not fully consider the causal relationship between entities in a document.

3 Methods

3.1 Overall Architecture

This paper defines ADE detection as a classification task. We develop the contextualized graph embedding model as illustrated in Figure 1. There are three components of the model. **(1) Graph Construction with Contextualized Embeddings.** We construct a heterogeneous graph to represent words and documents in the whole dataset, following TextGCN [39], and use pre-trained language models, specifically BERT [7] and its domain-specific variants, to obtain the contextualized text representation. **(2) Graph-based Text Encoding.** To capture neighborhood information in the heterogeneous graph, the feature matrix obtained from the embedding layer and the adjacency matrix from the constructed graph are fed into graph encoders. The feature embeddings are iteratively updated in the heterogeneous relational networks of words and documents. **(3) ADE Classification.** We follow the BertGCN model [22] to fuse contextualized embedding and graph networks with a weight coefficient to balance these two branches. Furthermore, we build an attentive classification layer to allow more critical content to contribute more to predictions. Fig. 1 shows the overall model architecture. The details of these components are introduced in the following sections.

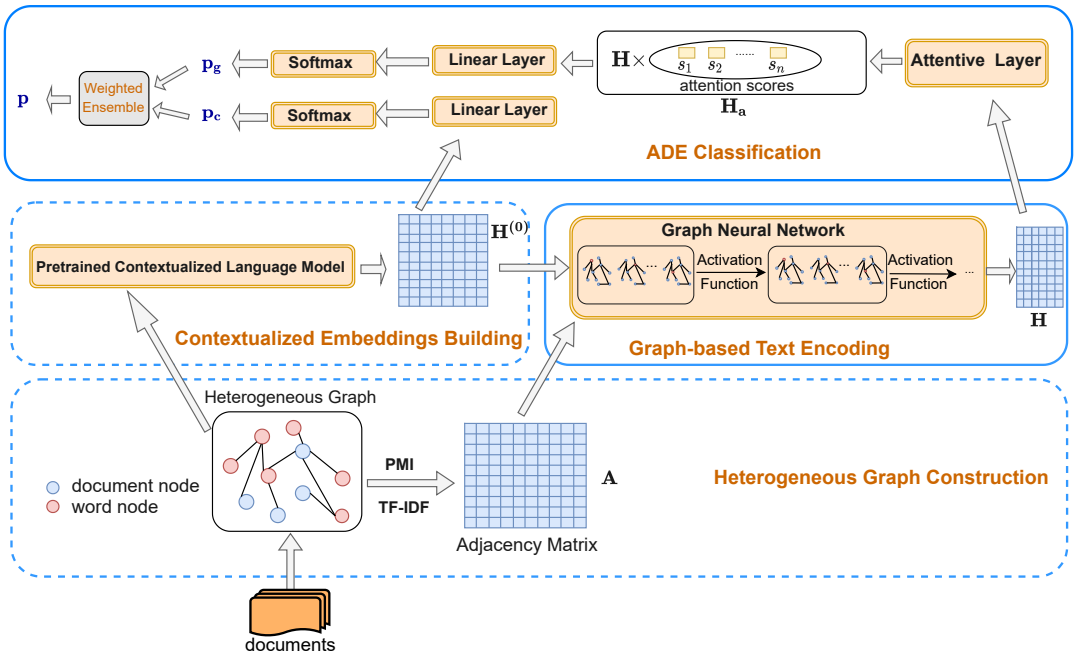


Figure 1: The illustration of the model architecture with contextualized graph embeddings for ADE detection

3.2 Graph Construction

3.2.1 Heterogeneous Graph

We first represent text as a graph before feeding it to neural networks. Representing text in a heterogeneous graph can provide different perspectives for text encoding and improve ADE detection. The process of graph construction follows TextGCN [39]. Nodes in the graph represent documents and different words. The number of nodes n equal to the number of documents n_d plus the number of unique words n_w in the whole dataset, i.e., $n = n_d + n_w$. There are two types of edges, i.e., word-word and document-word edges. We use the term frequency-inverse document frequency (TF-IDF) of one word in the document to represent the weight of a document-word edge, while the weight of a word-word edge is based on positive point-wise mutual information (PMI) of two words. We can represent the weight between the node i and the node j as:

$$\mathbf{A}_{ij} = \begin{cases} \text{PMI}(i,j), & \text{PMI} > 0; \text{i, j: words} \\ \text{TF-IDF}_{\text{ij}}, & \text{i: document, j: word} \\ 1, & i = j \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

3.2.2 Contextualized Embeddings

We used three pre-trained contextualized language models to obtain embeddings for documents. They are all BERT-based models but pre-trained with different strategies or corpora collected from different domains. The pre-trained language embeddings include:

- RoBERTa [23]: a pre-trained model with masked language modeling (MLM) objective on English language. In this paper, we used the base version.
- BioBERT [19]: a BERT-based model trained with biomedical corpora including PubMed abstracts and PubMed Central full-text articles.
- ClinicalBERT [2]: another domain-specific BERT-based model which is trained on clinical notes from the MIMIC-III database [14].

Given the dimension of embeddings denoted as d, the final output of contextualized text encoding are denoted as $\mathbf{H}_{doc} \in \mathbb{R}^{n_d \times d}$. We then apply a zero matrix as the initialization of word nodes to get the feature matrix input to GNN:

$$\mathbf{H}^{(\mathbf{0})} = \begin{pmatrix} \mathbf{H}_{doc} \\ \mathbf{0} \end{pmatrix} \tag{2}$$

where $\mathbf{H}^{(\mathbf{0})} \in \mathbb{R}^{(n_d+n_w)\times d}$.

3.3 Graph-based Text Encoding

This section employs a graph-based model for text encoding and capturing complex heterogeneous relationships. Graph neural networks are powerful models to mine and capture the relations and dependencies of graph data. Specifically, we apply two graph neural networks, i.e., Graph Convolutional Network (GCN) [18] and Graph Attention Network (GAT) [33], which are commonly used in different tasks. Graph convolution encodes the topological structure of the heterogeneous graph, enables label influence propagation, and achieves effective modeling of ADE corpora. In this section, we introduce their principles.

GCN is a category of Convolutional Graph Neural Networks (ConvGNNs) models. It is a spectral-based model which incorporates nodes' feature information from their neighbors. It can be seen as a multilayer neural network limited to undirected graphs where the number of layers is fixed. Each layer has different weights to better process cyclic mutual dependencies. GCN is the approximations and simplifications of Spectral CNN. It approximates spectral graph convolutions using convolutional architecture to get a localized first-order representation.

A graph G consists of nodes set V, and edge sets E. $\mathbf{A}$ is the adjacency matrix obtained from the step of graph construction, and $\hat{\mathbf{A}}$ is its normalized form. $\mathbf{D}$ is the degree matrix, where $\mathbf{D}_{ij} = \sum_j \mathbf{A}_{ij}$. In the GCN model, multiple layers are stacked to integrate information about higher-order neighborhoods. In the m-th layer, the feature matrix is updated as:

$$\mathbf{H}^{(m)} = f(\hat{\mathbf{A}}\mathbf{H}^{(m-1)}\mathbf{W}^{(m-1)}), \qquad \hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}, \tag{3}$$

where $\mathbf{H}^{(m)} \in \mathbb{R}^{n\times d_m}$,and $\mathbf{W}^{(m-1)} \in \mathbb{R}^{d_{m-1}\times d_m}$ is the weight matrix, $\mathbf{H}^{(0)}$ is the output from contextualized language models, and $f(\cdot)$ is an activation function.

Being similar to GCN, GAT is also a ConvGNNs model. However, it is spatial-based neural networks, where node information is propagated within edges and graph convolutions are finally decided by the spatial relation. It employs the message passing process and attention mechanism to learn relations between nodes. Graph attention layers in GAT assign different attention scores to one node's distant neighbors and prioritize the importance of different types of nodes.

3.4 Classification Layers

The GNN-based text encoding produces hidden feature representations $\mathbf{H} \in \mathbb{R}^{n\times d_c}$. We propose to use an attention mechanism (Eq. 4) to put more attention on nodes with more important information related to positive or negative ADE classes, denoted as

$$\mathbf{s} = \text{softmax}(\mathbf{w_a}\mathbf{H}^{\mathrm{T}}), \tag{4}$$

where $\mathbf{w_a} \in \mathbb{R}^{d_c}$ and $\mathbf{s} = (s_1, s_2, \cdots, s_n) \in \mathbb{R}^n$ is the attention weight vector containing attention score of each node. Attention scores from the attentive classification layer are different from the attention layer of GAT. Here, attention scores measure which nodes are more important to the graph, while in the attention layer of GAT, attention scores decide the importance of one node to the other node in the neighborhood. The weight is assigned to feature matrix to obtained attentive hidden representation weighted by attention scores, i.e., $\mathbf{H_a} = [s_1 \times \mathbf{h'_1}, s_2 \times \mathbf{h'_2}, ..., s_n \times \mathbf{h'_n}]$.

Then, we apply the softmax classifier over the graph-based encoding and obtain the probability of each class as:

$$\mathbf{p}_g = \text{softmax}(\mathbf{W_f}\mathbf{H_a}^{\mathrm{T}}\mathbf{v_f}), \tag{5}$$

where $\mathbf{W_f} \in \mathbb{R}^{n \times d_h}$ and $\mathbf{v_f} \in \mathbb{R}^{n \times 2}$ are the weight matrices. We apply the same calculation as Eq. 5 but with different weight matrices to pretrained contextualized embeddings $\mathbf{H^{(0)}}$. Finally, we get $\mathbf{p}_c$ as the prediction probability from the contextualized embeddings. A weight coefficient $\lambda \in [0, 1)$ is introduced to balance the result from graph-based encoding models and the result from BERT-based contextualized models:

$$\mathbf{p} = \lambda \mathbf{p}_g + (1 - \lambda)\mathbf{p}_c. \tag{6}$$

This weighted strategy can also be viewed as an ensemble of two classifiers or the interpolation of the prediction probability of two classifiers.

3.5 Model Training

We apply the negative log-likelihood loss function as the training objective. Because data in one of the datasets used in our study is imbalanced and the number of instances of this dataset is not large where the down-sampling method is not suitable, we use the weighted negative log-likelihood loss function to solve the data imbalance problem [30]. Assuming that the number of documents containing ADE is N_1 and the number of documents not containing ADE is N_2, the weight w_+ for documents predicted as positive samples is $\frac{N_2}{N_1+N_2}$ and the weight w_- for documents predicted as negative samples is $\frac{N_1}{N_1+N_2}$. The weighted loss function is:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(w_+ y_i \log(p_i) + w_-(1 - y_i)\log(1 - p_i)), \tag{7}$$

where N is the number of documents in one batch and y_i is the true label of a document. When a document contains ADE, y_i equals to 1; otherwise, y_i equals to 0. The Adam optimizer [17] is used for model optimization. To control the learning rate, we use the multiple-step learning rate scheduler. The learning rate scheduler decays the learning rate by the parameter γ when the number of epochs reaches a specific number.

4 Experiment

4.1 Data and Pre-processing

We used three datasets from the biomedical domain and social media to evaluate the performance of baselines and our model. The details of these datasets are shown in Table 1. We perform data pre-processing before building graph representation. Specifically, stop words, punctuation, and numbers are removed. For the data collected from Twitter, we use the tweet-preprocessor Python package [1] to remove URLs, emojis, and some reserved words for tweets.

Table 1: A statistical summary of datasets

Dataset	Documents	ADR	non-ADR
SMM4H	2418	1209	1209
TwiMed-Pub	1000	191	809
TwiMed-Twitter	625	232	393

TwiMed-Twitter and TwiMed-Pub [2] The TwiMed dataset [3] includes two sets collected from different domains, i.e., TwiMed-Twitter and TwiMed-Pub. They consist of documents from Twitter and PubMed, respectively. People with different backgrounds annotate diseases, symptoms, drugs, and their relations in each document. There are three types of relations: Outcome-negative, Outcome-positive, and Reason-to-use. When a document is annotated as outcome-negative, it is marked as ADE (positive). Otherwise, we mark it as non-ADE (negative). The TwiMed-Pub has a small number of documents containing ADEs. The weighted loss function is used to solve the issue of imbalanced classification. Models are evaluated by 10-fold cross-validation.
SMM4H dataset [3] [29, 24] The dataset is from Social Media Mining for Health Applications (#SMM4H) shared tasks. Documents collected from Twitter contain a description of drugs and diseases. The dataset contains

[1] https://pypi.org/project/tweet-preprocessor/
[2] https://github.com/nestoralvaro/TwiMed
[3] https://healthlanguageprocessing.org/smm4h-2021/task-1/

17,385 tweets for training and 915 tweets for testing. In our experiment, since this dataset is large enough, we conduct downsampling to mitigate the problem of imbalance, where we only use 2418 tweets, half of which are negative (non-ADE) and the other half are positive (ADE). The training tweets are split into train and validation sets, with a ratio of 9:1. We use the official validation set to evaluate the model performance for a fair comparison with baseline models developed in the SMM4H shared task, such as [40, 28, 15].

4.2 Baselines and Evaluation

Precision (P), Recall (R), and F1-score are commonly used to measure different models in a classification task. We report these three metrics in our results and mainly use the F1-score to compare models' performance in our experiments. We consider two sets of baseline models for performance comparison: 1) models explicitly designed for ADE detection and 2) pre-trained contextualized models.

Customized models for ADE detection include:

- CNN-Transfer [21] (CNN-T for short): a CNN-based model with transfer learning module. It has two sentence classifiers and a shared feature extractor based on CNN.
- HTR-MSA [35]: a model with hierarchical tweet representation and multi-head self-attention. This model learns word representations and tweet representations with CNN and Bi-LSTM. The multi-head self-attention mechanism is also applied.
- ATL [21]: a model based on adversarial transfer learning for the ADE detection, where corpus-shared features are exploited.
- MSAM [41]: a model with the multihop self-attention mechanism. It captures contextual information using Bi-LSTM and applies an attention mechanism in multiple steps to generate semantic representations of sentences.
- IAN [1]: interactive attention networks, a model to interactively learn attentions in the context and model targets and context separately.

We compare our model with pre-trained language models on the SMM4H dataset as it is a recent dataset not studied by the aforementioned ADE detection baselines. We use the base version of pretrained models in our experiments for a fair comparison, which is the same setting as in the compared baselines.

- BERT [7]: a language representation models pre-training with unlabeled text. Yaseen et al. [40] proposed a model that combined LSTM with a BERT encoder for ADE detection, denoted as BERT-LSTM in this paper.
- RoBERTa [23]: a BERT-based model on the English language with slightly different pre-training strategies. Pimpalkhute et al. [28] developed a data augmentation method with RoBERTa text encoder for ADE detection, denoted as RoBERTa-aug in this paper.
- BERTweet [25]: a domain-specific model for English Tweets with the same architecture as BERT-base. Kayastha et al. [15] built a model with BERTweet and single-layer BiLSTM for ADE detection, denoted as BERTweet-LSTM in this paper.

4.3 Experimental Setup

We use Python 3.7 and PyTorch 1.7.1 to implement the model. The hyper-parameters we tuned in our experiments are presented in Table 2. In our experiment, we set the hyper-parameter of the learning rate scheduler γ and the milestone of epoch number to 0.1 and 30, respectively.

Table 2: Choices of hyper-parameters

Hyper-parameters	Choices
Learning rate for text encoder	$2e^{-5}$, $3e^{-5}$, $1e^{-4}$
Learning rate for classifier	$1e^{-4}$, $5e^{-4}$, $1e^{-3}$
Learning rate for graph-based models	$1e^{-3}$, $3e^{-3}$, $5e^{-3}$
Hidden dimension for GNN	200, 300, 400
Weight coefficient λ	0, 0.1 0.3, 0.5, 0.7, 0.9

4.4 Main Results

We compared our model with baseline models for the ADE detection task to validate the performance of our model. Table 3 and Table 4 show the results of TwiMed and SMM4H dataset, respectively. Our model achieves the best performance for all datasets compared with other methods in terms of F1-score. The best result of TwiMed-Pub is obtained with ClinicalBERT embeddings and a GAT encoder. As for SMM4H and TwiMed-Twitter, the best combination of building blocks is RoBERTa embeddings and GCN encoder.

Table 3: Results of TwiMed datasets

Datasets	Metrics	HTR-MSA [35]	CNN-T [21]	MSAM [41]	IAN [1]	ATL [21]	Ours
TwiMed-Pub	P (%)	75.0	81.3	85.8	87.8	81.5	**88.4**
	R (%)	66.0	63.9	**85.2**	73.8	67.0	85.0
	F1 (%)	70.2	71.6	85.3	79.2	73.4	**86.7**
TwiMed-Twitter	P (%)	60.7	61.8	74.8	83.6	63.7	**84.2**
	R (%)	61.7	60.0	**85.6**	81.3	63.4	83.7
	F1 (%)	61.2	60.9	79.9	82.4	63.5	**83.9**

Table 4: Results of SMM4H dataset

Methods	P (%)	R (%)	F1 (%)
BERT-LSTM [40]	77.0	72.0	74.0
BERTweet-LSTM [15]	81.2	86.2	83.6
RoBERTa-aug [28]	82.1	85.7	84.3
Ours	**86.7**	**93.4**	**89.9**

As shown in Table 3, performances of HTR-MSA, ATL, and CNN-Transfer are lower than others. The network structures of these three models are complex, resulting in a large amount of data being required. Thus, it performs worse than other models on small corpora. MSAM achieves the best performance on recall, while our model performs the best on precision and F1-score. Our model can balance precision and recall better. The competitive performance on the three datasets also shows the high generalization ability of our model. In Table 3, the performances of most models on the two datasets are significantly different. It is challenging to detect ADEs from tweets since tweets are informal text and contain much colloquial language. However, our model performs well on the TwiMed-Twitter dataset, showing that it can effectively encode information from the informal text and better capture relationships of entities in a document. From Table 4, we can find that other models are all BERT-based models. In contrast, our model employs GNN architectures, which suggests GNN can significantly improve models' performance on this task.

4.5 Analyses and Discussion

We further analyze the contextualized graph embedding model in this section, discuss the choice of different building blocks, and conduct a case study.

4.5.1 Choice of Graph Encoders

Our experiment examines GCN and GAT to study which one is more suitable for the ADE detection task. We record the best result under different graph encoders. For both GCN and GAT, we obtain the best result from RoBERTa for the SMM4H dataset and TwiMed-Twitter. For TwiMed-Pub, the best result is obtained using ClinicalBERT. From Table 5, we can find the results from the two GNNs are similar, showing that they both performed well on this task.

4.5.2 Choice of Pretrained Embeddings

We examine three contextualized language models in our experiment. We record the best results with different language models. When using RoBERTa, the best results for the SMM4H dataset, TwiMed-Pub, and TwiMed-Twitter are from GCN, GAT, and GCN, respectively. When using ClinicalBERT, the best results for the SMM4H dataset and TwiMed-Pub are from GAT, and for TwiMed-Twitter, the best result is from GCN. When using BioBERT, the choice of GNNs for best results is the same as using ClinicalBERT.

Table 5: Comparison on the choices of graph encoders, i.e., GCN and GAT

Graph Encoder	SMM4H			TwiMed-Pub			TwiMed-Twitter		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
GCN	86.7	93.4	**89.9**	88.6	84.3	86.4	84.2	83.7	**83.9**
GAT	84.8	92.3	88.4	88.4	85.0	**86.7**	83.1	81.9	82.5

Table 6: The effect of contextualized text embeddings obtained pretrained from different domains

Pretrained Embeddings	SMM4H			TwiMed-Pub			TwiMed-Twitter		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
RoBERTa	86.7	93.4	**89.9**	88.2	84.3	86.2	84.2	83.7	**83.9**
ClinicalBERT	81.5	92.3	86.6	88.4	85.0	**86.7**	80.1	80.7	80.4
BioBERT	80.9	93.4	86.7	88.2	84.0	86.0	81.2	80.6	80.9

From Table 6, we can find that, for TwiMed-Pub, there is little difference among the three pre-trained language models. However, for the SMM4H dataset and TwiMed-Twitter, RoBERTa performs better than others. The SMM4H dataset and TwiMed-Twitter dataset contain documents with many non-medical terms, while ClinicalBERT and BioBERT are trained with many medical terms. Therefore, when there are insufficient medical terms in the text, ClinicalBERT and BioBERT are unsuitable. RoBERTa is a better choice for informal text for this task.

4.5.3 Ablation Study on the Attention Classifier

To examine the effect of the attention classifier, we conduct an ablation study in our experiment. We remove the attentive classification layer and check the performance change in F1 scores.

From Table 7, we can find that after removing the attentive classification layer, values of F1-scores get decreased for all three datasets. It suggests that the attentive classification layer can improve the model to prioritize information in the heterogeneous graph. More meaningful content, such as the description of symptoms and drugs, medical terms, and other relevant information related to ADEs, can contribute more to final predictions by employing attention mechanisms in the classification layer.

We also notice that F1 scores increase with the attentive classification layer, while precision scores for the SMM4H and TwiMed-Twitter datasets decrease. The documents of these two datasets are both from Twitter. Tweets are informal texts that do not follow the logical order, and their structures are unclear. They lack medical terms, and some content that seems not to be related to ADEs may also help determine whether a document contains ADEs or not. After applying the attentive classification layer, the model puts more attention to parts directly related to the description of symptoms, resulting in a tendency where a tweet is more easily to be predicted as a positive sample. Therefore, the precision value decreases after employing the attention classification layer. Besides, we can find that the F1 score on the SMM4H dataset decreases to a greater extent without an attentive classification layer. This dataset contains more documents compared to others. It suggests that the attentive classification layer works better for larger datasets. For small corpora, models with simpler architectures also perform well.

4.5.4 Effect of Weight Coefficient λ

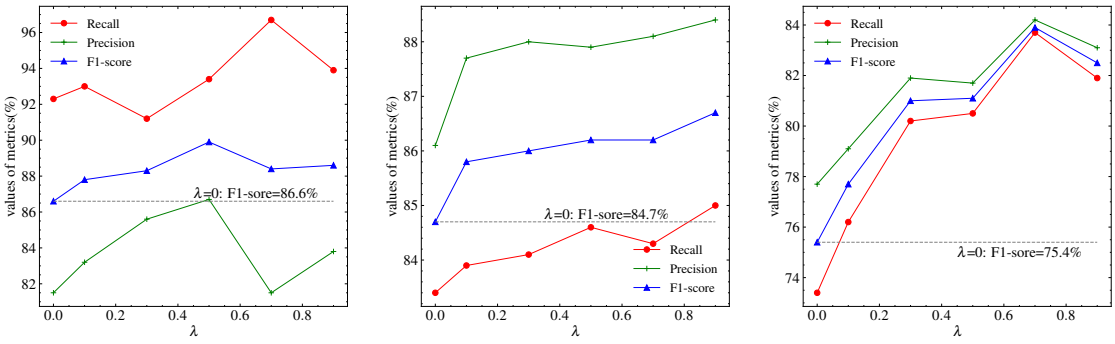
The weight coefficient λ's value controls the trade-off between the contextualized language models and graph neural networks. When λ equals zero, only BERT-based pre-trained contextualized embeddings are considered. In 2, dashed lines show the values of the F1-score when λ equals to zero. After employing GNNs ($\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$), we can find that the value of the F1-score increases on all three datasets. It demon-

Table 7: Comparison between our model and the model without attentive classification layer

	SMM4H			TwiMed-Pub			TwiMed-Twitter		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
our model	86.7	93.4	89.9	88.4	85.0	86.7	84.2	83.7	83.9
- attentive layer	87.0	90.1	88.5	87.8	83.9	85.8	84.6	82.2	83.3

strates that convolutional GNNs can improve the performance of our model significantly. Determining whether a symptom description is about the disease itself or adverse reactions resulting from the disease is a challenge in ADE detection. Utilizing GNNs helps solve this issue since GNNs can better capture the cause-effect relation and dependency between different entities of documents.

We can find the trend of the three lines are similar in respective plots of Figure 2. In terms of F1-score, the best choices of the value of λ for three datasets are 0.5 (SMM4H), 0.9 (TwiMed-Pub), and 0.7 (TwiMed-Twitter). It suggests how to choose the value of λ depending on which datasets we use and other model hyper-parameters. Also, when values of λ are greater than 0.5, the F1 scores are relatively high. Therefore, we can first choose a high value of λ to allow graph embeddings to contribute more.

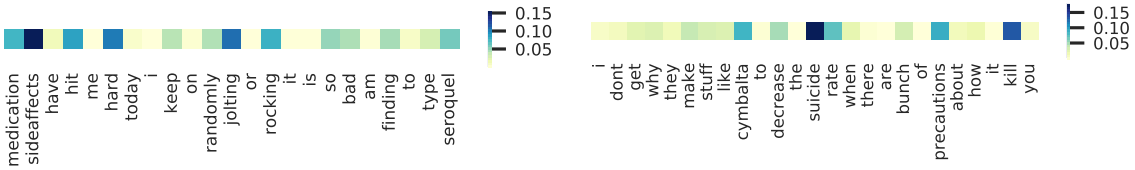


(a) SMM4H (b) TwiMed-Pub (c) TwiMed-Twitter

Figure 2: The effect of weight coefficient λ on values of metrics

4.5.5 Case Study

We conduct a case study to explore the effect of the attention mechanism in Eq. 4. We choose two documents classified as positive samples in the SMM4H test dataset, where one is classified correctly while the other one does not contain ADE. We record the attention scores of words of these two tweets and utilize a heap map to show the value of different words' attention scores in a document, illustrated in Figure 3. Figure 3a of a correctly classified tweet shows nouns (such as medication, sideaffects and seroquel), verbs (such as jolting), and sentiment words (such as hard and bad) related to drugs and symptoms get high attention scores. It helps the model put more attention on these important words. However, assigning high attention scores to such words does not ensure correct predictions. Figure 3b shows the attention scores of a tweet incorrectly classified as a positive sample. We can find that words related to symptoms, negative sentiment, and drugs are still getting high scores, while the tweet does not talk about ADE directly.



(a) A tweet with ADE (b) A tweet without ADE

Figure 3: Case study of the attention scores in two tweets: (a) with ADE; and (b) without ADE

5 Conclusion

The automated detection of adverse drug events from social content or biomedical literature requires the model to encode text information and capture the causal relation efficiently. This paper utilizes contextualized graph embeddings to learn contextual information and causal relations for ADE detection. We equip different convolutional graph neural networks with pre-trained language representation, develop an attention classifier to detect ADEs in documents and study the effects of different building components in our model. By comparing our model with other baseline methods, experiment results show that graph-based embeddings can better capture causal relationships and dependency between different entities in documents, leading to better detection performance.

Acknowledgment

We thank Professor Hongfei Lin for his kind support of this work. We acknowledge the computational resources provided by the Aalto Science-IT project and CSC - IT Center for Science, Finland. This work was supported by the Academy of Finland (grants 315896) and EU H2020 (grant 101016775).

References

[1] Ilseyar Alimova and Valery Solovyev. Interactive attention network for adverse drug reaction classification. In *Conference on Artificial Intelligence and Natural Language*, pages 185–196. Springer, 2018.

[2] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.

[3] Nestor Alvaro, Yusuke Miyao, and Nigel Collier. Twimed: Twitter and pubmed comparable corpus of drugs, diseases, symptoms, and their relations. *JMIR public health and surveillance*, 3(2):e6396, 2017.

[4] Danushka Bollegala, Richard Sloane, Simon Maskell, Joanna Hajne, and Munir Pirmohamed. Learning causality patterns for detecting adverse drug reactions from social media. *Journal of Medical Internet Research*, 2018.

[5] Anne Cocos, Alexander G Fiks, and Aaron J Masino. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *JAMIA*, 24(4):813–821, 2017.

[6] Bharath Dandala, Venkata Joopudi, and Murthy Devarakonda. Adverse drug events detection in clinical notes by jointly modeling entities and relations using neural networks. *Drug Safety*, 42(1):135–146, 2019.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[8] Molla S Donaldson, Janet M Corrigan, Linda T Kohn, et al. To err is human: building a safer health system, 2000.

[9] Lian Duan, Mohammad Khoshneshin, W Nick Street, and Mei Liu. Adverse drug effect detection. *IEEE Journal of Biomedical and Health Informatics*, 17(2):305–311, 2012.

[10] Suyu Ge, Tao Qi, Chuhan Wu, and Yongfeng Huang. Detecting and extracting of adverse drug reaction mentioning tweets with multi-head self attention. In *Proceedings of SMM4H Workshop*, pages 96–98, 2019.

[11] Trung Huynh, Yulan He, Alistair Willis, and Stefan Rüger. Adverse drug reaction classification with deep neural networks. In *COLING*, 2016.

[12] Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. Does the magic of bert apply to medical code assignment? a quantitative study. *Computers in Biology and Medicine*, 2021.

[13] Tianwen Jiang, Qingkai Zeng, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. Biomedical knowledge graphs construction from conditional statements. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(3):823–835, 2020.

[14] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[15] Tanay Kayastha, Pranjal Gupta, and Pushpak Bhattacharyya. BERT based Adverse Drug Effect Tweet Classification. In *Proceedings of SMM4H Workshop*, pages 88–90, 2021.

[16] Humayun Kayesh, Md Saiful Islam, and Junhu Wang. A causality driven approach to adverse drug reactions detection in tweets. In *International Conference on Advanced Data Mining and Applications*, pages 316–330. Springer, 2019.

[17] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[19] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[20] Hui Li, Xiao-Jing Guo, Xiao-Fei Ye, Hong Jiang, Wen-Min Du, Jin-Fang Xu, Xin-Ji Zhang, and Jia He. Adverse drug reactions of spontaneous reports in shanghai pediatric population. *PLoS One*, 9(2):e89829, 2014.

[21] Zhiheng Li, Zhihao Yang, Ling Luo, Yang Xiang, and Hongfei Lin. Exploiting adversarial transfer learning for adverse drug reaction detection from texts. *Journal of Biomedical Informatics*, 106:103431, 2020.

[22] Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. BertGCN: Transductive Text Classification by Combining GCN and BERT. *arXiv preprint arXiv:2105.05727*, 2021.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[24] Arjun Magge, Ari Klein, Antonio Miranda-Escalada, Mohammed Ali Al-Garadi, Ilseyar Alimova, Zulfat Miftahutdinov, Eulalia Farre, Salvador Lima-López, Ivan Flores, Karen O'Connor, et al. Overview of the sixth social media mining for health applications (# smm4h) shared tasks at naacl 2021. In *Proceedings of SMM4H Workshop*, pages 21–32, 2021.

[25] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.

[26] Azadeh Nikfarjam, Abeed Sarker, Karen O'connor, Rachel Ginn, and Graciela Gonzalez. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *JAMIA*, 22(3):671–681, 2015.

[27] Judea Pearl. *Causality*. Cambridge university press, 2009.

[28] Varad Pimpalkhute, Prajwal Nakhate, and Tausif Diwan. IIITN NLP at SMM4H 2021 Tasks: Transformer Models for Classification on Health-Related Imbalanced Twitter Datasets. In *Proceedings of SMM4H Workshop*, pages 118–122, 2021.

[29] Abeed Sarker and Graciela Gonzalez-Hernandez. Overview of the second social media mining for health (SMM4H) shared tasks at AMIA 2017. *Training*, 1(10,822):1239, 2017.

[30] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[31] Sunghwan Sohn, Cheryl Clark, Scott R Halgrim, Sean P Murphy, Christopher G Chute, and Hongfang Liu. Medxn: an open source medication extraction and normalization tool for clinical text. *JAMIA*, 21(5):858–865, 2014.

[32] Janet Sultana, Paola Cutroneo, and Gianluca Trifirò. Clinical and economic burden of adverse drug reactions. *Journal of Pharmacology & Pharmacotherapeutics*, 4(Suppl1):S73, 2013.

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[34] Qiang Wei, Zongcheng Ji, Zhiheng Li, Jingcheng Du, Jingqi Wang, Jun Xu, Yang Xiang, Firat Tiryaki, Stephen Wu, Yaoyun Zhang, et al. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *JAMIA*, 27(1):13–21, 2020.

[35] Chuhan Wu, Fangzhao Wu, Junxin Liu, Sixing Wu, Yongfeng Huang, and Xing Xie. Detecting tweets mentioning drug name and adverse drug reaction with hierarchical tweet representation and multi-head self-attention. In *Proceedings of SMM4H Workshop*, pages 34–37, 2018.

[36] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[37] Susmitha Wunnava, Xiao Qin, Tabassum Kakar, Xiangnan Kong, and Elke Rundensteiner. A dual-attention network for joint named entity recognition and sentence classification of adverse drug events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3414–3423, 2020.

[38] Hua Xu, Shane P Stenner, Son Doan, Kevin B Johnson, Lemuel R Waitman, and Joshua C Denny. Medex: a medication information extraction system for clinical narratives. *JAMIA*, 17(1):19–24, 2010.

[39] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of AAAI*, volume 33, pages 7370–7377, 2019.

[40] Usama Yaseen and Stefan Langer. Neural Text Classification and Stacked Heterogeneous Embeddings for Named Entity Recognition in SMM4H 2021. In *Proceedings of SMM4H Workshop*, pages 83–87, 2021.

[41] Tongxuan Zhang, Hongfei Lin, Yuqi Ren, Liang Yang, Bo Xu, Zhihao Yang, Jian Wang, and Yijia Zhang. Adverse drug reaction detection via a multihop self-attention mechanism. *BMC bioinformatics*, 20(1):1–11, 2019.

[42] Tongxuan Zhang, Hongfei Lin, Bo Xu, Yuqi Ren, Zhihao Yang, Jian Wang, and Xiaodong Duan. Gated iterative capsule network for adverse drug reaction detection from social media. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 387–390. IEEE, 2020.